

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 01-228039

(43)Date of publication of application : 12.09.1989

(51)Int.Cl.

G06F 12/10

G06F 12/08

(21)Application number : 63-319664

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 20.12.1988

(72)Inventor : BAUM RICHARD I
BORDEN TERRY L
BUTWELL JUSTIN R
CLARK CARL E
GANEK ALAN G
LUM JAMES
MALL MICHAEL G
PAGE DAVID R
PLAMBECK KENNETH E
SCALZI CASPER A
SCHMALZ RICHARD J

(30)Priority

Priority number : 88 154740

Priority date : 10.02.1988

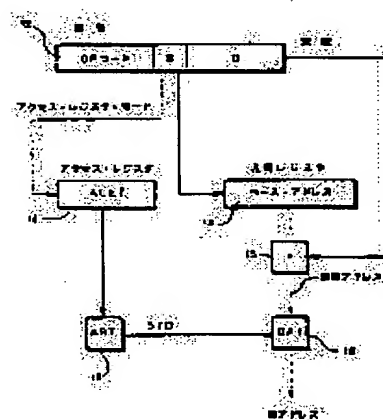
Priority country : US

(54) COMPUTER SYSTEM

(57)Abstract:

PURPOSE: To make permission hierarchical and non-hierarchical as well by providing a program permitting mechanism equipped with plural access registers and an access register transformation(ART) means.

CONSTITUTION: An instruction 12 contains an OP code, B field for designating a general-purpose register 14 containing a base address, and displacement D for generating the logical address of a storage area operand while being combined with the base address in the general-purpose register 14 through an adder 15. An ART operation 10 uses contents in access registers 16 for providing a segment table descriptor (STD) to be used for dynamic address translation(DAT). When an access list entry token (AELT) is transformed by the ART 10, the STD corresponding to an address space for storing data is applied. The real address of an operand to be used for a system is generated from DAT 18.



LEGAL STATUS

[Date of request for examination]

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑪ 公開特許公報(A) 平1-228039

⑫ Int. Cl.⁴

G 06 F 12/10
12/08

識別記号

庁内整理番号

H-7010-5B
Q-7010-5B

⑬ 公開 平成1年(1989)9月12日

審査請求 有 請求項の数 3 (全30頁)

⑭ 発明の名称 コンピュータ・システム

⑮ 特 願 昭63-319664

⑯ 出 願 昭63(1988)12月20日

優先権主張 ⑰ 1988年2月10日 ⑱ 米国(US) ⑲ 154740

⑳ 発 明 者 リチャード・イーワイ アメリカ合衆国ニューヨーク州ボキプシイ、アーバー・ヒル・ドライブ5番地

㉑ 発 明 者 テリイ・リイ・ボーデ アメリカ合衆国ニューヨーク州ボキプシイ、パートナーズ・テラス21番地

㉒ 出 願 人 インターナショナル・アメリカ合衆国10504、ニューヨーク州 アーモンク(番地なし)
ビジネス・マシーンズ・コーポレーション

㉓ 代 理 人 弁理士 頼 宮 孝一 外1名
最終頁に続く

明 細 書

1. 発明の名称 コンピュータ・システム

2. 特許請求の範囲

(1) 中央処理装置と、データ処理命令を有するプログラムを実行するために前記中央処理装置を動作させるオペレーティング・システムと、前記プログラムにより処理されるべきデータを含む複数のアドレス空間に割り振られる主記憶装置とを備えるコンピュータ・システムにおいて、

各々が前記アドレス空間のうちの1つを指定するアクセス・リスト・エントリ・トークンを含み、前記命令のうちの被選択命令のオペランドによってアドレス可能な複数のアクセス・レジスタと、

前記中央処理装置で実行中のプログラムにおける前記被選択命令のオペランドによって指定されたアクセス・リスト・エントリ・トークンが指定するアドレス空間へのアクセスを許可するための前記中央処理装置中のアクセス・レジスタ変換手段と、

を含むプログラム許可機構を設けたことを特徴とするコンピュータ・システム。

(2) 中央処理装置と、データ処理命令を有するプログラムを実行するために前記中央処理装置を動作させるオペレーティング・システムと、前記プログラムにより処理されるべきデータを含む複数のアドレス空間に割り振られる主記憶装置とを備えるコンピュータ・システムにおいて、

プログラム制御を第1アドレス空間から第2アドレス空間へ移動する制御移動手段と、

少なくとも指定されたプログラム許可を有するプログラムによってのみ前記第1及び第2のアドレス空間へのアクセスを提供する許可手段と、

を含むプログラム許可機構を設けたことを特徴とするコンピュータ・システム。

(3) 前記許可手段が、前記プログラム許可が階層的または非階層的となるように前記プログラム許可の値を大きくまたは小さくする許可変更手段を有する、特許請求の範囲第(2)項記載のコンピュータ・システム。

3. 発明の詳細な説明

A. 産業上の利用分野

本発明は、データ処理システムで実行中のプログラムが多重仮想アドレス空間への並行アクセスを有するようにそのプログラムに許可

(authorization) を提供する多重仮想システム(MVS)の許可機構に関する。特に、本発明は、その許可が非階層的である、すなわち被呼出しプログラムが呼出しプログラムに比べて同等かもしくはより高い許可権を有しなくても良いような許可機構に関する。

B. 従来技術

多重仮想アドレス空間での仮想アドレス指定を利用するデータ処理システムはよく知られている。このようなシステムの例として、MVS制御プログラミングを用いたIBMシステム/370がある。IBMシステム/370の構成については“IBM System/370-XA Principles of Operation”(資料番号SA22-7085-1)に詳しい。これに記載されているMVSシステムは、命令実

個のレジスタ・フィールドを有することにより複数の汎用レジスタをアドレス指定することを規定する。

CPUは、各々が32ビット位置を有する16個の制御レジスタを備えている。制御レジスタにおけるビット位置は、プログラム事象記録(PER)のようなシステムにおける特定機能に割り当てられたり、動作を条件付けるすなわち抑制するためまたはその機能に要求される特別な情報を供給するために用いられる。

米国特許第RE27251号明細書は、メモリの物理的ブロックに関連する4ビット・コード化記憶保護キーを開示している。その保護キーは、データへのアクセスを制御するために、プログラムに関連するPSWキーと比較される。

米国特許第4096573号明細書及び同第4136385号明細書もMVSシステムを開示している。それによれば、主記憶装置は複数のユーザにより使用されるアドレス空間として割り振られ、各アドレス空間はすべてのユーザに共通の部

分、割込み処理、タイミング機能、初期プログラム・ローディングその他の機械関連機能のための順序づけ及び処理機構を備えた中央処理装置(CPU)と、直接アドレス指定可能でCPUによる高速データ処理を可能にする主記憶装置とを含んでいる。主記憶装置はCPUと一体的に、または独立ユニットとして構成される。

機械中のレジスタにプログラム状況ワード(PSW)が位置している。このワードは、命令の順序づけを制御したりCPUの状態を決めたりするために用いられる命令アドレスその他の情報を含む。命令は、16個の汎用レジスタのうちの1個以上における情報を指し示すことができる。これらのレジスタは、アドレス演算における基底アドレス・レジスタ及び指標レジスタとして用いられる。汎用演算論理動作におけるアキュムレータとして用いられることもある。それらの汎用レジスタは、番号0ないし15によって識別され、命令における4ビット・レジスタ・フィールドによって指し示される。いくつかの命令は、数

分を含む。各ユーザは、プログラムやデータを自身に割り当てられたアドレス空間の私用部分に置いておくと、それらを他のユーザから分離することができ、共通部分におくと、他のユーザによるアクセスが可能となる。このようなシステムでは、2つのアドレス空間の間でデータを移動することができる。その場合、第1のアドレス空間にあるプログラムがデータをその私用域から共通域へ移し、ついで第2のアドレス空間にあるプログラムにデータを移動するよう知らせる。共通域を複数のアドレス空間の間の連絡域として使用すると、共通域の容量が増え、その分私用域の容量が減る。あるプログラムから別の制御プログラムへの通知はサブシステムあるいは制御プログラムだけになり得る。データは記憶保護キーにより保護されるが、その数は16しかなく、共通にアドレス指定され得る情報を他のサブシステムまたは許可されたプログラムによる不注目の書き込みから保護するのは不十分である。

米国特許第4355355号明細書は、汎用レ

ジスタ(GPR)に関連するアクセス・レジスタ(AR)を開示している。ARにはアドレス空間識別子、たとえば一意的なセグメント・テーブル記述子(STD)がロードされる。STDは、主記憶装置におけるセグメント・テーブル・アドレスとセグメント・テーブル長フィールドを含む。ARは所定数だけ設けられ、それぞれGPRと関連づけられて、ARの数までのデータ・アドレス空間のサブセットを定義する(各GPRについてのアドレス空間の数が2以上になることはない)。プロセッサにおいては、16個のARをそれぞれ16個のGPRに関連づけることができる。AR中のアドレス空間識別子たとえばSTDは、関連するGPRが記憶域オペランド・ベース・レジスタとして選択される時、たとえばGPRがIBMシステム/370命令のBフィールドによって選択される時に、アドレス変換のために選択される。各ARは、関連するAR中のSTDを用いてそのデータ・アドレス空間を定義する代わりに、プログラム・アドレス空間AR中のSTDを用いるよ

うに指定することもできる。しかし、関連するGPRが記憶域オペランド・ベース・レジスタ以外の目的、たとえばインデックス・レジスタまたはデータの転送元あるいは転送先レジスタとして選択されると、ARのアドレス空間識別子の内容たとえばSTDはアドレス変換のためには選択されない。実行プログラム・アドレス空間(データを含んでいてもよい)を定義し制御するのに16番目のARを設けることができる。

この米国特許に開示された発明は、また、実行中のプログラムによって関連するアドレス空間へのアクセスが許されるようなタイプのアクセスを制御するために、AR制御ベクトル(ARCV)レジスタを各ARにも関連させることにより、各プログラムの各アドレス空間へのアクセスについての許可権を含んでいる。ARまたはその関連するARCVにおける特別なフィールドによって、そのARの内容が使用可能かどうかを指示することが可能であり、そのためにそのフィールドは使用可能なアドレス空間識別子を含む。あるいは、

そのフィールドは、そのアドレス空間識別子がどこかよそで得られなければならないように使用禁止にされる。各ARと共にこの特別なフィールドによって、使用禁止ARのGPRがもう1つのARによって定められたアドレス空間に関連する基数値を特定することができる。すなわち、使用禁止ARのGPRが、AR0によって定められたプログラム・アドレス空間におけるデータについての基底アドレスを特定することができる。

米国特許第4430705号明細書は従来のMVSシステムの改良を開示している。二重アドレス空間(DAS)システムが米国特許第4388537号明細書及び同第4500952号明細書に開示されている。そのシステムでは、異なるアドレス空間に関連する許可権テーブルによって許可されるなら、問題プログラムがその異なるアドレス空間へのアドレス可能性を得ることができる。従来のMVSシステムに比べて、開示されたDASシステムは、制御レジスタ、汎用レジスタ及びPSWを使用する。PSWにおける1つのビット

が、DAS動作モードを指示するために割り当てられる。

DASは、準特権プログラムによって使用可能な一次アドレス空間及び二次アドレス空間の2つのアドレス空間を作る。情報を移動させるために3つの命令が付加される。MVCP命令は、データを二次アドレス空間から一次アドレス空間へ移動させ、MVCS命令は、データを一次アドレス空間から二次アドレス空間へ移動させ、そしてMVCK命令は、データを同じアドレス空間における異なる記憶保護域間で移動させる。CPUを一次モードにすることができる。このモードでは、命令及び論理的であるように定義されたそのオペランド・アドレスが一次アドレス空間を参照する。あるいは、CPUを二次モードにすることができる。このモードでは、論理的であるように定義されたそのオペランド・アドレスが二次アドレス空間を参照する。しかしながら、二次モードでは、命令が一次アドレス空間から取り出されるかまたは二次アドレス空間から取り出されるかは予想で

きない。したがって、二次モードで実行されるプログラムは、アドレス空間の共通部分に置かれる。この部分は、一次アドレス空間と二次アドレス空間との間で共用される。

DASによって、許可権の種々のレベルで動作するプログラムが監視プログラムを呼び込むことなく直接にリンクされ得る。プログラム呼出し(PC)及びプログラム移動(PT)の命令によって、制御の種々のレベルまたは同じレベルで動作するプログラムの中で制御を移動するための保護機構が提供される。あるアドレス空間におけるプログラムから別のアドレス空間におけるプログラムへのPC命令は、空間切替えを伴うプログラム呼出し(PC-ss)動作と呼ばれる。空間切替えを伴わない同じアドレス空間におけるプログラムへのPC命令は、現一次に対するプログラム呼出し(PC-cp)動作と呼ばれる。そのPC-cp及びPC-ssの両動作は、特権及び許可権のより高いレベルへの変更を考慮している。戻り機能がPT命令によって実行される。この命令は、

めに用いられる。

DASは、また、ASN第1テーブル及びASN第2テーブルを確立する。これらのテーブルは、エントリ・テーブルのエントリにおいて見出されたASN値を翻訳すなわち変換するために用いられる。アドレス空間に割り当てられるASNは、ASN第1テーブル指標(AFX)及びASN第2テーブル指標(ASX)から成る。AFXによってASN第1テーブルへの指定がなされ、ASN第1テーブルのエントリが指し示される。このエントリは、ASN第2テーブルの指定を含む。ASN第1テーブルのエントリが有効なら、ASX値によってASN第2テーブルへの指定がなされる。このようにして見出されたASN第2テーブルのエントリ(ASTE)は、セグメント・テーブルの起点(STO)を含む。この起点は、目標アドレス空間の実アドレスを決める動的アドレス変換機構(DAT)によって用いられる。

制御レジスタ3におけるPSWキー・マスク(PKM)のビットを、問題プログラム状態で用

現一次に対してであったり、または空間切替え動作(PT-ss)を伴ったりすることがある。

制御の移動を達成するために、DASは数個のテーブルを確立する。連係テーブルがプログラム呼出し動作によって使用のために確立される。連係テーブルの各エントリは、エントリ・テーブル・アドレスを含む。そのエントリ・テーブル・アドレスにおいてエントリ・テーブルが確立される。そのテーブルの各エントリは、呼び出されることになっているプログラムについてのエントリ情報を含む。各プログラム呼出し動作については、DASが、連係指標(LX)及びエントリ指標(EX)から成るPC番号を作成する。LXによって連係テーブルへの指定がなされ、エントリ・テーブル・アドレスを得る。またEXによってそのアドレスにおけるエントリ・テーブルへの指定がなされ、被呼出しプログラムについてのエントリ情報を得る。エントリ・テーブル・エントリ・データはアドレス空間番号(ASN)を含む。この番号は、各プログラムのアドレス空間を識別するた

いて、どのキー及びエントリ・ポイントがプログラムに対して許可されるのかを制御する。PKMは、プログラム呼出し(PC)及びプログラム移動(PT)の命令によって変更され、アドレス空間パラメータをロードする(LASP)命令によってロードされる。PKMを問題プログラム状態で用いて、PSWキー値を制御する。二次アクセス・キーを特定するMVCP、MVCS及びMVCK命令に対して有効なPSWキー値を制御するために、そしてPC命令によって呼び出すことができるエントリ・ポイントを制御するために、準特権命令によってPSWにPSWキー値を設定することができる。

プログラム呼出し(PC)命令では、PSWキー・マスクがエントリ・テーブルのエントリにおける許可キー・マスク(AKM)とAND演算される。その結果が非ゼロなら、プログラムはPC命令を発行するのを許可される。エントリ・テーブルのエントリは、エントリ・キー・マスク(EKM)を含む。このマスクは、追加キーを含む。

これらの追加キーに対して被呼出しプログラムは許可される。PCルーチンが制御を受け取ったときには、EKMは制御レジスタ3のPKMへOR演算される。このようにして、PKMによって提供される許可権を増大させることができる。

プログラム移動(PT)命令では、制御レジスタ3中のPKMがPSWキー・マスクとAND演算される。このPSWキー・マスクはオペランドとして特定される。その演算結果が制御レジスタ3中のPKMに取って代わる。このようにして、PKMによって提供される許可権を低減させることができる。

MVCP、MVCS及びMVCKの命令では、一次アドレス空間へのアクセスがPC動作におけるPSWキーによって許可される。MVCP及びMVCS命令の場合には、二次空間をアクセスするのに用いられるべきアクセス・キーをオペランドが特定する。問題プログラム状態では、オペランドの二次空間アクセス・キーに対応する制御レジスタ3中のPKMビットが1である場合にのみ、

もとで走っているプログラムの許可権を指し示す。ASN許可には監視プログラム状態プログラムと問題プログラムとの両方が必要である。そのATに対してチェックされるべきAXは制御レジスタ4中に位置する。そして、そのアドレス空間についての許可権テーブルの起点(AT0)は、目標アドレス空間のASTB中に位置する。許可権テーブルの各エントリは、PビットとSビットの2個のビットから成る。AT中のそのAXエントリに対応するAXを有するプログラムは、その対応するPビットが1ならそのアドレス空間をその一次アドレス空間として確立するために許されるし、またその対応するSビットが1ならそのアドレス空間をその二次アドレス空間として確立するために許される。

DAS機能の使用にはいくつかの制限がある。MVCP及びMVCSの命令は、データを一次アドレス空間と二次アドレス空間との間で移動させることができるだけである。DASは、一次アドレス空間または二次アドレス空間でない2つのい

MVCPまたはMVCS動作が実行される。MVCK命令の場合には、原始データ領域をアクセスするのに用いられるべきアクセス・キーをオペランドが特定する。問題プログラム状態では、オペランドの原始データ領域アクセス・キーに対応する制御レジスタ3中のPKMビットが1である場合にのみ、MVCK動作が実行される。

ASN許可は、許可指標(AX)を用いて特定のアドレス空間を確立するために、現許可指標に関連したプログラムが許されるかどうかをテストするプロセスである。そのASN許可は、PT-ss動作の1部分としてまたは空間切替えを伴う二次ASN設定(SSAR-ss)動作として行なわれる。そのSSAR-ss動作は、二次アドレス空間を所望アドレス空間に設定する。各アドレス空間はその許可権テーブル(AT)と関連している。この許可権テーブルは用いられるあらゆるAXに対して1個のエントリを含む。ATにおけるAXのエントリは、アドレス空間に対するPT及びSSAR命令を発行するためにそのAXの

いずれかのアドレス空間の間でデータを移動させることはできない。アドレス空間で実行されているすべてのプログラムは、そのアドレス空間に関連する許可指標を用いる。プログラムによる二次モードの使用は、そのプログラムが共通領域に存在することを必要とする。一次アドレス空間及び二次アドレス空間の両方におけるデータをアクセスするために、一次モードと二次モードとの間で頻繁に切替えを行なうことは、性能を著しく低下させる。PC命令は階層型の連係のみを実行する。監視プログラム状態のプログラムから問題プログラム状態のプログラムへ制御を与えるためにPC命令を用いるのは実用的でない。なぜなら、戻すためにPT命令を用いることはできないからである。PC命令は、その許可権を増大させることによりPSWキー・マスクを替えることができるだけである。空間切替えPC命令は、常に呼出しプログラムのアドレス空間に対する被呼出しプログラムのアクセスを与える。最後に、PC命令は、PSWキーを替えない。それで、取出し防止されたコー

ドを呼び出すことはできない。

米国特許第4037214号明細書は、アクセス・キー・レジスタ(AKR)にある複数のアクセス・キーが命令アドレス、シンク・オペランド・アドレス及びソース・オペランド・アドレスに基づいて記憶域アクセスのアドレス空間を切り替える水平アドレス指定システムを開示している。

米国特許第4521848号明細書は、データを非特権状態でアクセスできるクロスメモリ構成において複数の仮想アドレス空間へのアクセスを制御するための別の機構を開示している。

米国特許第3787813号明細書は、ケイパビリティ・レジスタを用いたデータ処理装置を開示している。このデータ処理装置は中央処理装置及び記憶装置を有し、記憶装置中の情報はセグメント化され、中央処理装置に設けられる複数のケイパビリティ・レジスタはそれぞれ情報セグメントのベース・アドレス及び限界アドレスを示す記述子情報を記憶する。ケイパビリティ・レジスタの1つは、中央処理装置で現在実行中のプログラ

ムに関連するセグメント・ポインタ・テーブルを含む情報セグメントのベース・アドレス及び限界アドレスを定義する情報を保持し、別の1つは、記憶装置中の各情報セグメントに対するエントリを有する主ケイパビリティ・テーブルを含む情報セグメントのベース・アドレス及び限界アドレスを定義する情報を保持する。セグメント・ポインタ・テーブルは、主セグメント・テーブルの異なるエントリを定義するためのポインタとして使用されるデータ・ワードのリストを含む。

米国特許第4366536号明細書は、別々に記憶されていてアセンブリ・レベルのコマンドからなる複数のデータ処理手順を選択して連係し、さらに複数の可変データ域のうちの1つを選択するディジタル計算機システムを開示している。このシステムは、データ処理手順、可変データ域及び連係アドレスを記憶するメモリと、データ処理手順を記憶しているメモリをアクセスするためのプログラム・カウンタと、データ及び連係アドレスを記憶しているメモリをアクセスするためのレ

ジスタと、選択されたデータ処理手順に含まれるアセンブリ・レベルのコマンドを実行中のデータ処理手順に含まれるアセンブリ・レベルのコマンド及び前に選択されたアドレスに従って実行するハードウェア装置とを含む。

米国特許第4288903号明細書は、スタック域を制御するためのスタック制御レジスタ・グループを開示している。スタック機構中に形成されるユーザ・プログラムによって直接制御されるデータ・スタック域の開始アドレスは、データ・スタック・ポインタ・レジスタに保持される。

米国特許第4454580号明細書は、ある論理アドレス空間にあるプログラムから新しい論理アドレス空間にある別のプログラムへ実行を移す方法を開示している。呼出しプログラムは被呼出しプログラムに対する選択的なセグメント割振りを制御し、被呼出しプログラムは割り振られるセグメントの長さを制御する。したがって、同じプログラムが繰り返し呼び出されても、他のプログラム又は前の呼出しにおける同じプログラムの機

能またはデータは影響を受けない。また、データ・セグメントの割振りを実行まで延ばすことができ、それにより、他のプログラムの詳細を知らずに書かれたプログラムの実行に融通性を持たせることができる。

米国特許第4297743号明細書は、それぞれが異なる特権レベルを表わす複数のリングの階層構成を開示している。特権レベルの低いリングへの分岐が許され、特権レベルは読取専用状況については異なってもよいが、読取書込状況についてはそれは許されない。該特許は3つの記憶域、すなわち変数を記憶するための作業域、レジスタの内容を保管するための保管域、及び手順間でパラメータを移動するための連絡域を有するスタック・フレームを教示する。ユーザは手順呼出しの前に、保管すべきレジスタを指定しなければならず、また被呼出し手順へ渡すべきパラメータを連絡域へロードしなければならない。システムは一連のスタック・フレームにおいて呼出しを記録しており、それにより戻りが可能になる。

米国特許第4044334号明細書は、アドレス可能空間の複数のセグメントのうちの1つでデータベース・レコードを見出すためのデータベース・ポインタを検索するシステムを開示している。

1982年1月刊のIBM Technical Disclosure Bulletin 第24巻、第8号の4401~4403頁に掲載されている "Method of Revoking a Capability Containing a Pointer-Type Identifier without Accessing the Capability" と題する論文は、アドレス空間番号 (ASN) をアドレス空間ケイバリティのためのポインタ・タイプ識別子として使用する技術を開示している。これは二重アドレス空間機構に関するものであり、アクセスが有効かどうかはアドレス変換に関連するASN第2テーブル・エントリ (ASTE) を用いて決定することができるので、アクセスの有効性を調べるためにアドレス空間を設定してはならない。一般に、ケイバリティによるオブジェクトへのアクセスは、ケイバリティ中の固有コードとオブジェクトが等しい場合にのみ許される。

プログラムの一次アドレス空間、DUALまたはPSALを参照するかどうかを決めるために、そしてそのような参照が所与のEAXに対して許可されるかどうかを決めるために、アドレス空間の許可レベルをテストすることができるテスト動作を提供することが含まれる。

D. 課題を解決するための手段

本発明の多重アドレス空間 (MAS) ファシリティは、16個の32ビット・アクセス・レジスタ0ないし15を提供する。DATがオンでPSWビット16及び17が2進の01であるときには、記憶域オペランドの論理アドレスを特定するのに用いられる命令のBまたはRフィールドが、汎用レジスタばかりでなくアクセス・レジスタをも指し示す。その指し示された汎用レジスタが、記憶域オペランドの論理アドレスを形成するために通常の方法で用いられる。DATにより用いられるセグメント・テーブル指定を特定してアドレス空間についての論理アドレスを翻訳すなわち変

換するにより、論理アドレスが関係するアドレス空間を特定するために、その指し示されたアクセス・レジスタが用いられる。アクセス・レジスタ自身は、そのセグメント・テーブル指定を含まない。

C. 発明が解決しようとする課題

本発明の目的は、プログラムに対してアドレス空間から命令を取り出したり1つ以上の他のいずれかのアドレス空間にオペランドを取り出す、すなわち記憶したりすることを許可するアーキテクチャを提供することである。

本発明の目的には、許可が所望すれば階層的にも非階層的にもなり得るような許可機構を提供することが含まれる。

また、本発明の目的には、アドレス空間をアクセスする能力が、プログラム呼出し順にすべてのプログラムに対して与えられたり、許可の特定レベルのみを有するプログラムに限定されたり、またはいくつかの異なる許可レベルに対して許されたりするような非階層的許可機構を提供することが含まれる。

さらに、本発明の目的には、ALETが呼出し

換することにより、論理アドレスが関係するアドレス空間を特定するために、その指し示されたアクセス・レジスタが用いられる。アクセス・レジスタ自身は、そのセグメント・テーブル指定を含まない。

アクセス・レジスタは、制御レジスタ1または7にそれぞれ記憶されている一次または二次のセグメント・テーブル指定を特定することができるし、またはASTEに含まれるセグメント・テーブル指定 (STD) を特定することができる。後者の場合には、アクセス・レジスタは、アクセス・リストと呼ばれるテーブル中のエントリをアクセス・リスト・エントリ・トークン (ALET) によって指し示す。そして、指し示されたアクセス・リストのエントリがそのASTEを指し示す。

アクセス・リストが、指名可能ユニット・アクセス・リスト (DUAL) または一次空間アクセス・リスト (PSAL) であることもある。アクセス・リストは指名可能なユニットまたは一次アドレス空間のいずれかに関連しているが、そのリ

スト中の有効なエントリは、指名可能なユニットの作業を行なうためにある順番に実行される種々のプログラムに関連するようになされている。また、各プログラムが特定の許可権を有することができるようになされている。この許可権によって、そのプログラムに関連しているアクセス・リストのエントリのみを用いることが許される。その許可権は、制御レジスタ8中の18ビット拡張許可指標(EAX)によって表わされる。関係する許可機構において用いられる他の構成要素は、アクセス・リストのエントリにおける私用ビット、アクセス・リストのエントリにおけるアクセス・リスト・エントリ許可指標(ALEAX)、そしてDASファシリティにより提供される許可権テーブルのSビットである。

アクセス・リストのエントリにおける私用ビット及びALEAXフィールドは、ALEによって表わされるアドレス空間に対するプログラムのアクセスを許可したりまたは禁止したりする高性能の許可機構を提供する。その私用ビットが0とな

グラム構造を確立することができる。EAX変更、PSWキー、PSWキー・マスク及び二次アドレス空間の変更についての許可権に関するオプションは、異なった許可権を異なったプログラムまたは同じ被呼出しプログラムと関連づける手段を与える。各プログラムの許可権は、制御プログラムによって管理されるエントリ・テーブルで規定される。同じプログラムを異なったPC番号で呼出せるようにエントリ・テーブルを設定することにより、どのPC番号を呼出しに用いるかに応じて異なった許可権をプログラムに割り当てることができる。エントリ・テーブルは、プログラムがどのPC番号を用いて他のプログラムを呼び出せるかということの制御も可能にする。

連係スタック機構によって提供されるスタッキング・プログラム呼出し及びプログラム戻りの連係オペレーションは、異なったアドレス空間にあって異なったレベルの許可権を持つプログラムを連係することができる。実行状態ならびに汎用レジスタ及びアクセス・レジスタの内容はスタッキン

ると、アクセス・リストを用いて実行するすべてのプログラムがALEによって表わされるアドレス空間をアクセスすることができる。また、そのALE私用ビットが1となり、そして制御レジスタ8におけるユーザのEAXが、ALEAXフィールドに等しくなると、特定のEAXを伴うプログラムはALEによって表わされるアドレス空間をアクセスすることができる。最後に、ALE私用ビットが1となり、そしてユーザの制御レジスタ8のEAXが、1に等しいSビットを有する目標空間の許可権テーブルにおけるエントリを選択すると、異なるEAXで走っている複数のプログラムがALEによって表わされるアドレス空間をアクセスすることができる。

連係スタック機構は、任意の異なった許可権レベルで動いているプログラムが制御プログラムの介入なしに直接連係されるのを可能にする。呼出しプログラム及び被呼出しプログラムのシーケンスにおける各プログラムの許可権の程度は任意に異なってもよく、したがって非階層的なプロ

グ・プログラム呼出し命令の実行の間保管され、プログラム戻り命令の実行の間に部分的に復元される。連係スタックは、連係オペレーションにおいて実行状態及びレジスタ内容の保管及び復元を効率的に行なう手段を提供する。

E. 実施例

本発明の多重アドレス空間(MAS)機構は、二重アドレス空間(DAS)機構及びアクセス・レジスタ・システムを改良したものである。MAS機構は、DAS機構との互換性を有しかつそれに追加する形で動くように設計されており、若干の変更及び改良はあるが、ほとんどの部分でDAS機構のものと同一テーブル及びレジスタ配置を使用する。アクセス・レジスタ変換(ART)システムは、ユーザがアクセス・レジスタ・システムを十分に利用できるようにすると共に、機械のアドレス指定機能をユーザから分離し保護する。ARTルックアサイド・バッファ(ALB)の使用はARTの性能を高める。

サービス提供者(プログラム)は普通、ユーザ

に使用させたいデータまたはプログラムを含むアドレス空間を1以上所有する。サービス提供者は、プログラムにプログラム呼出し(PC)番号を割り当てることによって、ユーザに対してそれらのプログラムを使用可能にする。PC番号割当て操作は、プログラム制御を移すためのリンクを確立すること、サービス呼出し者(プログラム)に必要とされる許可特性を指定すること、及びサービス提供者のプログラムの許可特性を割り当てることを含む。プログラム制御はあるアドレス空間から別のアドレス空間へ移すことができ、また同じアドレス空間に留まってもよい。いずれにしても、それに伴って許可があるレベルから別のレベルへ変えられることがあり、その結果、より大きな、より小さな、または異なった許可が与えられる。サービス提供者は呼出し者のレベルとは異なった許可レベルでランすることができ、それによりサービス提供者ルーチンは呼出し者がアクセスできないアドレス空間にあるデータをアクセスすることができる。ユーザ及びサービス提供者

は、アクセス・リストにおいて私用アドレス空間として指示されていないすべての空間をアクセスすることができる。さらにサービス提供者はユーザがアクセスできない選択されたアドレス空間をアクセスすることができる。同様に、サービス提供者に対し、ユーザがアクセスできる選択されたアドレス空間へのアクセスを禁止することができる。

プログラム命令の実行は2つのオペレーションに分けると都合がよい。最初のオペレーションは、実行すべき命令の取出しである。2番目のオペレーションはオペランドのアドレス指定であり、命令実行中に処理されるデータを取り出したり記憶したりするために行なわれる。ARモードのMASでは、命令は一次アドレス空間として設定されたアドレス空間から取り出される。一次アドレス空間の設定には空間切替え操作が必要な場合もある。

第1図は、本発明に従うアクセス・レジスタを用いてオペランドをアドレス指定する様子を示している。動的アドレス変換で使用するSTDを得

るためにアクセス・レジスタの内容を使用するプロセスをアクセス・レジスタ変換(ART)オペレーションと呼び、第1図では参照番号10で示されている。命令12は、OPコード、ベース・アドレスを含む汎用レジスタ14を指定するBフィールド、及び加算器15で汎用レジスタ14のベース・アドレスと組み合わせられて記憶域オペランドの論理アドレスを生成する変位Dを含んでいる。アクセス・レジスタ・モードにおいては、BフィールドはALETを含むアクセス・レジスタ16も指定する。ALETは、ART10で変換されると、データを記憶するアドレス空間に対するSTDを与える。ART10からのSTDは加算器15からの論理アドレスと組み合わせることができ、動的アドレス変換(DAT)18と一緒に変換されると、システムによって使用されるオペランドの実アドレスが生成される。命令12は、図示のBフィールド及び変位Dの他に、記憶域オペランドの論理アドレスを含む汎用レジスタを指定するRフィールドを含んでいてもよい。

本発明に従うアクセス・レジスタの使用は、次の移動(MVC)命令で説明することもできる。

MVC 0(L, 1), 0(2)

この命令は、長さLの第2オペランドを第1オペランド記憶位置に移動する。第2オペランドの論理アドレスは汎用レジスタ2にあり、第1オペランド記憶位置の論理アドレスは汎用レジスタ1にある。第2オペランドを含むアドレス空間はアクセス・レジスタ2にあるALETにより指定され、第1オペランドのアドレス空間はアクセス・レジスタ1にあるALETにより指定される。これら2つのアドレス空間は異なってもよく、また現命令のアドレス空間と異なってもよい。

ALETを用いた実アドレスへの変換の概要を第2図に示す。20は0番から15番までの汎用レジスタのアレイを示す。22は同じく0番から15番までのアクセス・レジスタのアレイを示し、各アクセス・レジスタはアレイ20の対応する汎用レジスタと対になっている。ALET中のアクセス・リスト・エントリ番号(ALEN)がアク

セス・リスト24または25の1つのエントリを選択する。アクセス・リスト24はDUAL(指名可能ユニット・アクセス・リスト)であり、アクセス・リスト25はPSAL(一次空間アクセス・リスト)である。第2図の例では、アクセス・レジスタ2のALENは、PSAL25のエントリ3を指し示す。DUAL24の起点は指名可能ユニット・アクセス・リスト記述子(DUALD)28により指定され、PSAL25の起点は一次空間アクセス・リスト記述子(PSALD)27により指定される。あとで述べるように、DUALD28は制御レジスタ2(CR2)にあるエントリを解放することにより見い出され、PSALD27は制御レジスタ5(CR5)にあるエントリを解放することにより見い出される。ARTで使用するアクセス・リスト記述子は有効アクセス・リスト記述子(ALD)として知られている。

アクセス・リストの各エントリは、ASN第2テーブル・エントリ(ASTE)98を指し示すASTEアドレスを含むが、ASTEはASN第

ために永続的に指名可能ユニットと関連づけられる。システムの指名可能ユニットごとに固有のDUALがある。DUALは、たとえ関連する指名可能ユニットが多くの異なるアドレス空間にあるプログラムを実行できるとしても、変わることはない。PSAL定義域は一次アドレス空間と関連づけられる。一次アドレス空間で実行されるすべてのプログラムは、そのアドレス空間のPSALを共用する。これにより、一次アドレス空間で実行されるプログラムは、共通の組のアドレス空間へのアクセスを共用することができる。一次アドレス空間がたとえば空間切替えPCオペレーションによって変わると、PSALも変わる。有効ALETを所有するユーザはDUAL24またはPSAL25のアクセス・リスト・エントリをアクセスできる。アクセスされたエントリは所望のアドレス空間を指定する。ALET及び選択された制御レジスタを用いて同様に制御される他の定義域を設けることも可能である。たとえば、システムにおけるすべてのプログラムのケイパビリティ

2テーブル(AST)30にあることもないこともある。実際にAST30に入っているものとはまったく独立に、ARTのためにASTEを生成してその機能を実行させることができる。ただし、PCで使用されるASTEはAST30になければならない。各ASTEはDAS機構で使用されるものと同様であり、DAT18で実アドレスを決定するためのSTD値を含む。

プログラムに対しては、それぞれが異なるケイパビリティ定義域を表わす2つのアクセス・リストが同時に使用可能である。一方のアクセス・リストは指名可能ユニット・アクセス・リスト(DUAL)と呼ばれ、他方は一次空間アクセス・リスト(PSAL)と呼ばれる。ALETのALENがいずれのアクセス・リストのエントリを指し示しているかは、ALET中の1ビットにより指定される。プログラムは、アクセス・リスト24及び25の各エントリを使用することができる。

DUAL定義域は、指名可能ユニット(タスクまたはプロセス)により実行されるプログラムの

ティを有するシステム・ワイド・アクセス・リスト(SWAL)を生成することができる。既存の機構をさらに改良するため、SASN定義域アクセス・リスト(SSAL)のごとき種々の定義域のサブセットを構成することもできる。

ARモードにおいては、ALETが一次アドレス空間及び二次アドレス空間にあるオペランドをそれぞれアクセスするように予約されているので、DUAL24のエントリ0及び1は使用されない。CPUのアドレス指定モードは、後述のように、PSW中のビットによって指定される。CPUがARアドレス指定モードにあれば、0(X'00000000')のALETは常に一次アドレス空間を示し、1(X'00000001')のALETは常に二次アドレス空間を示す。第2図のブロック28はこれらの特別なALETを識別し、DAT18へのPASN及びSASNに対する正しいSTDを与える。CPUがホーム・アドレス指定モードにあれば、ホーム・アドレス空間が実行すべき命令及びデータのソースである。ホーム・

アドレス空間は、実行中のプログラムに対する監視プログラム制御情報を有するアドレス空間として定義される。オペレーティング・システムは、データ・アクセスの目的で各ホーム空間に対して2(X'00000002')のALENを割り当て、ホーム空間のためのSTDはARTによって得られる。第2図のブロック28に示されるように、ALENが0に等しいなら、STDが制御レジスタ1から得られるし、ALENが1に等しいなら、STDが制御レジスタ7から得られる。一次アドレス空間及び二次アドレス空間に対するSTD値はそれぞれ制御レジスタ1及び7に保持されるので(第3図参照)、アクセス・リスト・エントリ0及び1は使用されない。第2図の例では、PSAL25のエントリ0、1及び2は使用されず、無効エントリとしてマークされる。

ARTロックアサイド・バッファ(ALB)199は、ARTから生ずるSTDを保持するため、ARアレイ22、PSAL25及びAST30からの入力を受け取り保管する。ALB199はア

使用する。制御レジスタ1は一次セグメント・テーブル記述子(PSTD)を含む。そのビット1~19は一次セグメント・テーブル起点(PSTO)を指定し、ビット25~31は一次セグメント・テーブル長(PSTL)を指定する。制御レジスタ2のビット1~25は、後述のように、DUALDを見つけるためにMAS機構によって使用される指名可能ユニット制御テーブル起点(DUCTO)を指定する。制御レジスタ5のビット1~25は、一次ASTE起点(PASTE0)を指定する。後述のように、制御レジスタ5の内容は、PSAL起点と、一次アドレス空間についてのASTE中の他の情報を見つけるべくASTEエントリを指し示す。

制御レジスタ7は、ビット1~19が二次セグメント・テーブル起点(SSTO)を指定し、ビット25~31が二次セグメント・テーブル長(SSTL)を指定する二次セグメント・テーブル記述子(SSTD)を含む。制御レジスタ8のビット0~15は、本発明のMAS機構で使用される並

クセス・リスト記述子DUALDまたはPSALDも保持する。同じALETが再び使用される場合、ALB199は正しい出力を直接DAT18へ供給し、ARTを繰り返す必要はない。

命令実行中のプログラムの制御及びCPUの状態に関する情報を与える制御レジスタ及びPSWをそれぞれ第3図及び第4図に示す。

第3図は、本発明のMAS機構に関する制御レジスタ0~15の内容を示している。これらの内容の多くは公知のDAS機構のものと同じであり、したがって、以下では主としてMAS機構のためになされた変更箇所について述べることにする。

制御レジスタ0のビット15(M)が1であれば、CPUがMASモードで動作していること、及び制御プログラムがMASをサポートすることを示す。MAS機構は、エントリ・テーブル・エントリ、ASN第2テーブル・エントリ、連係スタックの使用可能性、及びアクセス・レジスタ・モードに入る能力について新しいフォーマットを

要許可指標(EAX)を含む。後述のように、EAXはサービス提供者の制御のもとにエントリ・テーブル・エントリ中のビット内容の指定に仕掛けて変えることができ、これにより、プログラムによるアドレス空間のアクセスの許可を変更することができる。

制御レジスタ13は、ビット1~19がホーム・セグメント・テーブル起点(HSTD)を指定し、ビット25~31がホーム・セグメント・テーブル長(HSTL)を指定するホーム・セグメント・テーブル記述子(HSTD)を含む。制御レジスタ15のビット1~28は、後述のように、最後の連係スタック・オペレーションで定義された連係スタック・エントリのアドレスを含む。

第4図に示したPSWのフォーマットにおいて、ビット5はDAT18が活動状態かどうかを示すDATモード・ビット(T)である。ビット16及び17はアドレス指定モードを指定する。DAT18が活動状態の場合、ビット18及び17の組合せは、CPUが一次モード(00)、二次モー

F(10)、アクセス・レジスタ・モード(01)、及びホーム・モード(11)のいずれにあるかを示す。PSWのビット32は、ビット33～63の命令アドレスのフォーマットを指定するアドレス指定モード・ビットである。PSWの残りのフィールドの機能及びフォーマットはIBMシステム/370で公知のものである。

第2図のところで説明したALETのフォーマットを第5図に示す。ALETのビット7は一次リスト・ビットであり、これが1であれば、ALENがPSALを参照することを示す。ビット7が0であれば、ALENはDUALを参照する。ビット16～31はALENを含む。ALENを16倍すると、その積は、有効アクセス・リストの開始点から指定されたアクセス・リスト・エントリまでのバイト数に等しい。ARTの間にもしALENが有効アクセス・リスト外のエントリを指定するか、またはALETの左から7ビット(ビット0～6)がオールゼロでなければ、例外が認識される。ALENが有効ALDのアクセス・リス

ト長(ALL)によって決定されるアクセス・リストの終わりを越えたアドレスを指し示していると、アクセス・リスト・エントリは有効アクセス・リストの外にある。ALLについては第14図を参照されたい。ALETがX'00000000'またはX'00000001'(X'は16進表記を意味する)の場合は、上述のALETのフォーマットは適用されない。これら2つのALET値はARTプロセスによって特別の意味を割り当てられている。

ALETは、アクセス・レジスタ、汎用レジスタ及び記憶装置のいずれにも保持することができ、ユーザの問題プログラムによる操作からは保護されない。どのプログラムも命令によってALETの値をアクセス・レジスタ、汎用レジスタ及び記憶装置の間で自由に転送することができる。呼び出されたプログラムは、アクセス・レジスタの内容を自身が使用できる任意の記憶域に保管すること、アクセス・レジスタを自身の目的のためにロードし使用すること、及び呼出しプログラムへ戻る

前にアクセス・レジスタの元の内容を復元することが可能である。ALETのビット8～15はアクセス・リスト・エントリ順序番号(ALESN)を含む。ALETは問題プログラムからは保護されず、ユーザが誤ってその内容を別の値に変えてしまう可能性があるため、信頼性を高めるためにALESNをALETに含ませている。これはARTの間に検査される。

アクセス・リスト・エントリ(ALE)のフォーマットを第6図に示す。ALEのビット0は、ALEが無効かどうかを示す無効ビットである。ビット7は私用ビットで、もし0であれば、どのプログラムもARTオペレーションでこのアクセス・リスト・エントリを使用することができる。ビット7が1であれば、呼出しプログラムがこのアクセス・リスト・エントリを使用できるかどうかを決定するため、ビット16～31のアクセス・リスト拡張許可指標(ALEX)が使用される。ALEはビット8～15にALESN値を含む。これは、有効性検査のために、当該ALEを指定

したALETのALESNと比較される。ALEのビット65～89は、関連するアドレス空間の対応するASTEアドレスを含む。ALEのビット96～128は、後述のASTEエントリに関する有効性検査で使用されるASTE順序番号(ASTESN)を含む。

第2図に示したアクセス・リスト24及び25のエントリは、問題プログラムによる直接操作から保護するため、制御プログラムによって与えられる。この保護は、キー制御型の保護、またはアクセス・リストを問題プログラムがDATでアクセスできない実記憶域に置くことにより実現できる。ALEが有効か無効かはビット0によって示される。有効ALEは、適切に許可されたプログラムがアクセスできるアドレス空間を指定する。無効ALEは、割振り又は再割振りの場合は有効エントリとして使用できる。制御プログラムは、有効ALEを割り振ったり、以前に割り振ったALEを無効にしたりするサービスを提供する。

ALEの割振りは次のようなステップで行なわ

られる。まず、問題プログラムが、アドレス空間の識別子及びDUAL24またはPSAL25を指定する標識、すなわちALETの一次リスト・ビット7を制御プログラムに渡す。制御プログラムはこれに回答して、問題プログラムがアドレス空間のアクセスを許されているかどうかを検査する。もし許されていると、制御プログラムは指定されたアクセス・リストから無効エントリを1つ選択し、それを有効エントリに変え、その中にASTEアドレス及びASTESNを含ませることにより当該アドレス空間を指定し、割り振られたALEを指定するALETの値を問題プログラムに戻す。かくして、問題プログラムは新しいALETをアクセス・レジスタに置くことにより、アドレス空間をアクセスできるようになる。割り振られたALEはあとで制御プログラムの無効化サービスにより、無効にされることがある。

無効にされたALEが再割り振りされる場合、前の割り振りで指定されたアドレス空間とは別のアドレス空間が指定される。ユーザが概念的に正しく

ASTEを使用することができたすべてのプログラムまたは指名可能ユニットを保持する必要はない。かくして、ASTESNを変えることでアクセス権限を変えることができ、適切なASTESNなしにASTEを使用しようとする、例外又は割込みが生じる。これにより、オペレーティング・システムは、ASTESNの変更前にALEで許可されたペリパリティでASTEをアクセスしようとする試みを知ることができる。かくして、オペレーティング・システムは、新しいあるいは異なった空間のためにASTEを安全に再使用する、または既存の空間を使用するためその空間の現アクセス者の権限を再び有効化する機構を有する。

第7図、第8図及び第10図はそれぞれ連係テーブル、エントリ・テーブル及び連係スタックのエントリのフォーマットを示している。これらのテーブルは、同じまたは異なったアドレス空間にあるプログラムの間で制御を移すために連係を確立するためにMASファシリティによって使用される。

ないアドレス空間を指定するALETを誤って使用するのを避けるために、ALET及びALEの両方にALESNが保持される。制御プログラムは、ALEを割り振る時、同じALESNをALE及び指定されたALETの両方に置き、ALETを問題プログラムに戻す。ALEを再割り振りする時は、制御プログラムは再割り振りされるALEの中のALESNの値を変え、それが前に指定されたALET中のALESNの値と一致しないようにする。

ALEのASTESNについては、あとでASTEに関連してさらに詳しく説明するが、ここで重要なのは、ALE及びASTEに含まれるASTESNの値を比較することによって、ASTEの指定に関するALEの権限が確認されるということである。かくして、ASTEの再割当てが可能になり、このASTEを参照していたすべてのALEを調べる必要なしに、異なったASTESNを割り当ててその使用を制御することができる。ASTESNの使用により、制御プログラムは、

前述のように、PC番号は、システムによって呼び込まれかつサービス提供者によって構成される特定のPCルーチンを識別する。PCルーチンを提供する各サービス提供者は、自身のルーチンを定義するための1以上のエントリ・テーブルを所有する。エントリ・テーブルは、PCルーチンへのアクセスを要求するアドレス空間の連係テーブルに接続される。エントリ・テーブルの各エントリは1つのPCルーチンを定義し、その入口点及び動作特性を含むと共に、PC命令がスタッキングPCかどうかを示す。第7図は連係テーブル・エントリ(LET)のフォーマットを示したもので、無効ビット(I)、エントリ・テーブル起点(ETO)及びエントリ・テーブル長(ETL)を含み、これらによりエントリ・テーブル記述子が定義される。

第8図は第7図の連係テーブル・エントリにより指し示されるエントリ・テーブルのエントリのフォーマットを示している。エントリ・テーブル・エントリのビット0~15は許可キー・マスク

(AKM)を含み、これは、PC命令を出したプログラムが問題状態においてこのエントリ点を呼び出すことを許されているかどうかを検査するのに用いられる。ビット18~31は、PC-ss及びPC-cpのいずれが生じるかを示すエントリ・アドレス空間番号(EASN)を含む。EASNがオールゼロであれば、PC-cpが指定され、さもなければPC-ssが指定される。後者の場合、EASNは一次ASN(PASN)を置き換えるアドレス空間番号(ASN)を識別する。ビット32はアドレス指定モード・ビットであり、PCオペレーションでPSW中のアドレス指定モード・ビットを置き換える。ビット33~82に含まれるエントリ命令アドレス(EIA)は、PCオペレーションでPSW中の命令アドレスを置き換える。ビット63はエントリ問題状態ビットであり、PCオペレーションで現PSWの問題状態ビット(ビット15)を置き換える。ビット84~95は、PCオペレーションで汎用レジスタ4に置かれるエントリ・パラメータを含む。ビット98~

トリEAXがスタッキングPCオペレーションで制御レジスタ8にある現EAXを置き換えることを示す。Eビットが0であれば、制御レジスタ8にある現EAXは変更されない。ビット134はアドレス空間制御ビット(C)であり、これが1であれば、現PSWのビット17がスタッキングPCオペレーションで1にセットされることを示す。Cビットが0であれば、現PSWのビット17も0にセットされる。スタッキングPC命令が出される時は、CPUは一次空間モードまたはアクセス・レジスタ・モードのいずれかになければならないので、Cビットが1であれば、CPUはアクセス・レジスタ・モードになり、Cビットが0であれば、一次空間モードになる。ビット135は二次ASN制御ビット(S)であり、これが1であれば、スタッキングPC-ssオペレーションでビット18~31のEASNが新しい二次ASNになりかつ新しい二次セグメント・テーブル記述子(SSTD)が新しい一次セグメント・テーブル記述子(PSTD)に等しく設定されること

111は、後述のMビットの値に応じて汎用レジスタ3の内容とオア結合されるかまたはその内容を置き換えるエントリ・キー・マスク(EKM)を含む。ビット128はPCタイプ・ビット(T)で、これが1であれば、PC命令がスタッキング・オペレーションを遂行することを示す。ビット131はPSWキー制御ビット(K)であり、これが1であれば、ビット38~139のエントリ・キー(EK)がスタッキングPCオペレーションでPSW中のPSWキーを置き換えることを示す。このKビットが0であれば、PSWキーは変更されない。ビット132はPSWキー・マスク制御ビット(M)であり、これが1であれば、EKMがスタッキングPCオペレーションで汎用レジスタ3にあるPSWキー・マスクを置き換えることを示す。Mビットが0であれば、EKMはスタッキングPCオペレーションで制御レジスタ3にあるPSWキー・マスクとオア結合される。ビット133は拡張許可指標制御ビット(E)であり、これが1であれば、ビット144~159のエン

を示す。Sビットが0であれば、新しい二次アドレス空間番号(SASN)及びSSSTDは、呼出しプログラムの古い一次アドレス空間番号(PASN)及びPSTDにそれぞれ等しく設定される。EASNがオールゼロでなければ、ビット161~185のASTEアドレスはその右側に8個のゼロを付加されて、EASNのASN変換を適用した結果の実ASTEアドレスを形成する。かくして、エントリ・テーブル・エントリ中のEASN及びASTEアドレスがSTDを含むAST30のエントリを指し示すことがわかる。EASNのASN変換がASTEアドレスを得るために遂行されるかどうか、あるいはビット161~185のASTEアドレスがその指定されたASTEを見つけるのに使用されるかどうかは予測できない。CPUは性能改善のために、ビット161~185のASTEアドレスを用いてその指定されたASTEを見つけることができる。

第9図は、指名可能ユニットごとに制御プログラムによって作成され得る連係スタック35を示

している。この連係スタックは、スタッキング・オペレーションの間、実行状態と汎用レジスタ及びアクセス・レジスタの内容とを保管するのに用いる。連係スタックは、戻りオペレーションの間に保管内容の一部を復元するのに使用される。連係スタックは仮想記憶域にあり、関連する指名可能ユニットのためのホーム・アドレス空間に置かれる。第3図の制御レジスタに関連して説明したように、ホーム・アドレス空間は制御レジスタ13のHSTDによって指定される。これは連係スタック情報を保護し、ユーザ・アドレス空間で障害が発生した場合に連係スタック情報の回復を可能にする。

連係スタックは問題状態プログラムからは保護されており、それらのプログラムは、特別の抽出及び修正命令による場合を除いて、連係スタックに保管されている情報を調べたり修正したりすることはできない。連係スタック35は、正方向ポインタ及び逆方向ポインタによって連鎖される複数の連係スタック・セクション38、37及び3

タの内容(ARs)、PSWキー・マスク(PKM)、二次アドレス空間番号(SASN)、制御レジスタ8からのEAX、一次アドレス空間番号(PASN)、及びPSWの内容を含む。これらはすべてスタッキング命令の開始時のものである。さらに、使用されるPC番号も含まれる。後述するブランチ・スタック命令の場合は、PC番号の代わりにアドレス指定モード・ビット及びブランチ・アドレスが保管される。

連係スタック・エントリはいずれのタイプも8バイトの倍数の長さを有する。見出しエントリ及び後書きエントリはそれぞれ18バイトである。状態エントリは168バイトである(第10図の下側の数字はバイト番号を示している)。いずれのタイプもその終りに8バイトのエントリ記述子(第10図の連係スタック状態エントリの46参照)を有する。

エントリ記述子のビット0はアンスタック抑止ビット(U)である。見出しエントリまたは状態エントリのエントリ記述子においてビットUが1

8からなっているてもよい。

連係スタックには3種類のエントリ、すなわち逆方向ポインタを有する見出しエントリ40、正方向ポインタを有する後書きエントリ42、及び状態エントリ43(連係スタック・セクション38参照)がある。見出しエントリ及び後書きエントリはそれぞれ連係スタック・セクションの始め及び終りにあり、連係スタック・セクションを連鎖するのに使用される。見出しエントリ及び後書きエントリは制御プログラムにより作成され、スタッキング・オペレーションで実行状態及びレジスタ内容を保管すべく状態エントリが付加される。制御レジスタ15にある連係スタック・エントリ・アドレスは現状態エントリ44を指し示すか、またはセクション中の最後の状態エントリが除去されていると現セクションの見出しエントリを指し示す。

第10図は連係スタック状態エントリの内容を示したもので、スタッキングPC命令の場合、汎用レジスタの内容(GRs)、アクセス・レジス

であれば、プログラム戻りででのアンスタッキング・プロセスの間にスタック・オペレーション例外が認識される。状態エントリがスタッキング・プロセスの間に作成される時、そのエントリ記述子のビットUは0にセットされる。

エントリ記述子のビット1~7はエントリ・タイプ(ET)・コードで、当該エントリ記述子を含む連係スタック・エントリのタイプを指定する。

ETコードには次の4種類がある。

0000001	見出しエントリ
0000010	後書きエントリ
0000100	ブランチ状態エントリ
0000101	プログラム呼出し状態エントリ

エントリ記述子のビット8~15は制御プログラムによって与えられるセクション識別子(SI)である。スタッキング・プロセスで作成されるエントリでは、プロセスはSIを先行の連係スタック・エントリのSIに等しく設定する。エントリ記述子のビット16~31は、当該エントリの終

りから同じ連係スタック・セクション中の後書きエントリの始めまでのバイト数を指定する残余自由空間 (RFS) フィールドを形成する。エントリ記述子のビット32~47は、同じ連係スタック・セクションにおける次の連係スタック・エントリ(後書きエントリを除く)のサイズをバイトで指定する次エントリ・サイズ (NES) フィールドを形成する。

スタッキング・プロセスの間に連係スタックで新しい状態エントリを作成する場合、現連係スタック・セクションが新しいエントリを含むのに十分な自由空間を持っていると、新しいエントリは現連係スタック・エントリのエントリ記述子の直後に置かれる。現セクションに十分な自由空間がなくても、現セクションの後に別のセクションがあることを現セクションの後書きエントリが示していると、その後続セクションに十分な残余自由空間があれば、新しいエントリは後続セクションの見出しエントリのエントリ記述子の直後に置かれる。後続セクションがないことを後書きエントリ

が示していると、例外が認識され、プログラム割込みが起こる。その場合、制御プログラムが別のセクションを割り振り、それを現セクションに連鎖して、スタッキング・オペレーションを再実行させる。後続セクションがあっても、その中に十分な自由空間がなければ、例外が認識される。

スタッキング・オペレーションで新しい状態エントリ44がうまく作成されると、制御レジスタ15にある連係スタック・エントリ・アドレスは、新しいエントリのエントリ記述子の一番左のバイトを指定するように更新され、かくしてこのエントリが新しい現連係スタック・エントリになる。スタッキング・プロセスで状態エントリを作成する場合、そのNESフィールドにはゼロが置かれ、作成された状態エントリの長さは先行エントリのNESフィールドに置かれる。戻りオペレーションでは、汎用レジスタ及びアクセス・レジスタの内容、ならびに制御レジスタの種々の内容が連係スタック状態エントリ44から復元され、制御レジスタ15にある連係スタック・エントリ・アド

レスは前の連係スタック・エントリを指し示すように変更される。状態エントリが戻りオペレーションのアンスタッキング・プロセスの間に論理的に削除されると、先行エントリのNESフィールドにゼロが置かれる。

上述から明らかなように、連係スタックを使用すると、プログラム制御が戻り命令によって被呼出しプログラムから戻される時、呼出しプログラムの動作環境及び許可レベルを復元させることができる。連係オペレーションは開始点まで遡ると共に、無効な変更が生じないようユーザに要求する。

制御プログラムは、プログラムを含む各アドレス空間に対してアドレス空間番号 (ASN) を割り当てる。DAS機構に関連して述べたように、ASNはPC-ssオペレーションの間に交換されることがある。しかし、ASTEアドレスがETEにあるので(第8図参照)、PC-ssオペレーションにおいて、ASTEへのアクセスをASN変換なしにETEを用いて直接行なうことが

できる。制御プログラムは、アドレス空間に関連するASTEにポインタを置くことによって、STD、AT及び連係テーブルを各ASNに関連づける。それらのアドレス空間にあるデータも、ASTEを指し示すアクセス・リスト・エントリを制御プログラムに作成させることにより、アクセスすることができる。アドレス空間によってはデータしか含まないものもある。MASファシリティのケイバビリティがそのようなアドレス空間の作成を定める。そのようなアドレス空間はASNを持たない。データだけの空間の場合は、ASTE、STD、AT及びALEだけが使用される。

第11図及び第12図はそれぞれASN第1テーブル及びASN第2テーブルのエントリのフォーマットを示したものであるが、これらは前述のDAS機構のものときわめてよく似ている。第11図及び第12図の各エントリは1つのアドレス空間を表わし、そのアドレス空間への連係及びアドレス可能性を与えるべく制御プログラムによって設定される。

第12図はASTEのフォーマットを示している。ASTEのビット0は、ASTEが無効かどうかを示す無効ビットである。許可権テーブル起点(ATO)及び許可権テーブル長(ATL)は、関連する許可権テーブルの許可権テーブル記述子(ATD)を示す。ビット98~127は関連する連係テーブル記述子(LTD)を含み、ビット128~159は関連するアクセス・リスト記述子(ALD)を含む。ビット180~191はASTE順序番号(ASTESN)を含む。ASTEは、制御プログラムによるアドレス空間の作成及び削除に伴って再割り振りされることがあるので、新しく作成された各ASTEは新しい固有のASTESNを割り当てられる。ARTオペレーションを実行する時、ASTEが異なったアドレス空間または異なった権限に対して安全に再使用できるように、有効性検査として、アクセス・リスト中のASTESNとASTE中のASTESNが比較される。

第13図は、各ASTEに関連する許可権テ

ーブルを示している。DAS機構と同じく、各許可権テーブル・エントリはPビット及びSビットを1つずつ持っている。許可権テーブルのエントリは、関連するアドレス空間をアクセスするために、使用されているEAXの各値に対して許可権テーブルに1つのエントリが存在するように指標づけされる。後述のように、制御レジスタ8(CR8)にあるEAXの値に対応する許可権テーブル・エントリを用いて、プログラムがASTEに関連するアドレス空間のアクセスを許されているかどうかを調べることができる。

第14図は、制御レジスタ2にアドレスが保持されている指名可能ユニット制御テーブル(DUCT)のフォーマットを示している。DUCTのバイト16~19は指名可能ユニット・アクセス・リスト記述子(DUALD)を含む。DUCTの他のバイトはMAS機構では使用されないため、説明は省略する。

プログラム呼出し(PC)命令は、MASファシリティにおける連係機構の機能を上げるべく改

善された。PC命令が出された時、ETE(第8図)のTビット、すなわちビット128が1であれば、スタッキングPCオペレーションが実行される。スタッキングPCは、エントリ・テーブル・エントリ中の許可キー・マスクにより、エントリ・テーブルから入ることを許される。特に、空間切替えを伴うスタッキングPCは、(新しいプログラムに関連する)新しいEAXを制御レジスタ8に置くことがある。スタッキングPCは、汎用レジスタ0~15の内容、アクセス・レジスタ0~15の内容、更新された命令アドレス(戻りアドレス)を含む完全なPSW、PASN、SASN、PKM、EAX、エントリがプログラム呼出しによって作成されたことを示す指標、及び2ワード分の修正可能域をエントリに保管する。MASファシリティは、連係スタックの最後の状態エントリにおける情報を取り出してそのエントリにおける修正可能領域の内容を修正するための命令を含む。修正可能域の目的は、プログラムの障害が生じた時に適切な回復動作がとれるように、プログラム

の進行状況を記録できるようにすることにある。連係機能を改善するため2つの新しい命令、すなわちブランチ・スタック及びプログラム戻りが追加された。

ブランチ・スタック命令はPSW中の命令アドレスを変更し、第9図の連係スタックにブランチ状態エントリと呼ばれる状態エントリを作成する。ブランチ状態エントリは、ブランチ・スタック命令により作成されかつPC番号の代わりにブランチ・アドレスを含むということを除くと、プログラム呼出し状態エントリと同じである。ブランチ・スタック命令は、呼出しプログラムの中、または被呼出しプログラムの入口点(もしくはその近く)で使用する事ができる。入口点でのブランチ・スタック命令は、古い呼出しプログラムを変更することなく連係スタックを使用することを可能にする。

プログラム戻り命令は、スタッキングPC命令またはブランチ・スタック命令によって制御権を与えられたプログラムから戻るのに用いられる。

プログラム戻り命令は最後の連係スタック状態エントリを論理的に削除する。このエントリはプログラム呼出し状態エントリでもブランチ状態エントリでもよい。最後の状態エントリがプログラム呼出し状態エントリであれば、プログラム戻り命令は、エントリ中に保管されていたすべての状態情報ならびに汎用レジスタ2～14及びアクセス・レジスタ2～14の内容を復元する。0番、1番及び15番の汎用レジスタ及びアクセス・レジスタは、プログラム戻り命令によっては変更されない。最後の状態エントリがブランチ状態エントリであれば、プログラム戻り命令は、完全なPSW（PERマスク・ビットRを除く）ならびに汎用レジスタ2～14及びアクセス・レジスタ2～14の内容だけを復元する。呼び出されたプログラムが実行されていた間に発生したかもしれないPERの使用可能状態または使用禁止状態を妨げることのないように、PERマスク・ビットR（第4図参照）はプログラム戻りオペレーションによっては復元されない。スタッキング・プログラム呼

出し及びプログラム戻りの組合せは、非階層的なプログラム連係、すなわちある範囲の権限を持ったプログラムから、より小さい、より大きいまたはまったく異なる権限を持ったプログラムへの連係を可能にする。

第15図、第16図及び第17図は、スタッキングPCオペレーションを実行するのに必要なステップの論理フローを示している。この論理フローは、DASのプログラム呼出し命令の実行にも使用できる。図面中の説明は、種々の値が数学的にどのように操作されてアドレスを生成するかを述べている。第3図に戻って、制御レジスタ0のビット15（CR0.15）が0であれば、ETEは16バイトであり、DASプログラム呼出しオペレーションだけを実行できる。CR0.15が1であれば、ETEは32バイトであり、ETEビット128がDAS PC及びスタッキングPCのいずれを実行するかを示す。

第15図は、プログラム呼出しのPC番号変換オペレーションの論理フローを示している。CR

0.15=1であれば、制御レジスタ5にあるPASTE0（第3図参照）によって指定されるASTEが取り出される。この一次ASTEはビット86～127にLTDを含む（第12図参照）。プログラム呼出しがDASプログラム呼出しであれば（CR0.15=0）、通常のDASオペレーションと同じく、LTDは制御レジスタ5にある。プログラム呼出し命令50は、DASの場合と同様に、LX51及びEX52からなるPC番号を指定する。LX51は、加算操作54により連係テーブル起点（LTD）53と結合され、連係テーブル・エントリ55の実アドレスを生成する。EX52は、加算操作56により連係テーブル・エントリ55のエントリ・テーブル起点（ETO）と結合され、エントリ・テーブルにおけるエントリ・テーブル・エントリ（ETE）57の実アドレスを生成する。

第16図は、現一次に対するスタッキング・プログラム呼出し（PC-cp）及び空間切替を伴うスタッキング・プログラム呼出し（PC-

ss）を実行するために第15図から続く論理フローを示している。前述のように、ETE57のTビット80が1であれば、スタッキング・オペレーションが実行される。まず、AKM82の値が、問題状態におけるプログラム呼出し命令の実行前に制御レジスタ3に存在していたPKM84と83でアンド結合される。83でのアンド操作の結果がオールゼロであれば、プログラム呼出し命令はこの点から入ることを許されず、オペレーションは終了する。83でのアンド操作でいずれかのビットが一致していると、プログラムはこのエントリのところでプログラム呼出しを行なうことを許され、オペレーションが続けられる。プログラム呼出しが許されると、その実行前に存在していたPSW65、EAX86、PKM84、SASN88及びPASN89が連係スタック（LS）に置かれる。図には示していないが、汎用レジスタの内容、アクセス・レジスタの内容及びPC番号も連係スタックに置かれる（第10図参照）。アドレス指定モード・ビットA及びエント

リ命令アドレス (EIA) はそれぞれ PSW の 70 及び 71 のところに置かれる。ETE57 の P ビット及び C ビットはそれぞれ PSW の 72 及び 73 のところに置かれる。K ビットが 1 であれば、ETE57 の エントリ・キー (EK) は PSW の キー・フィールド 74 に置かれる。E ビットが 1 であれば、エントリ EAX (EEAX) は 75 に示すように制御レジスタ 8 に置かれる。エントリ・パラメータ (EP) は 76 に示すように汎用レジスタ 4 に置かれる。ETE57 の M ビットが 1 であれば、エントリ・キー・マスク (EKM) は 77 に示すように制御レジスタ 3 にある PKM を置き換える。しかし、M ビットが 0 であれば、EKM はオア操作 78 により制御レジスタ 3 の PKM とオア結合される。PC-cp オペレーションが実行されるか、またはスタッキング PC-ss が実行されかつ S ビットが 0 であれば、制御レジスタ 4 にあった PASN69 が制御レジスタ 3 の SASN79 を置き換え、制御レジスタ 1 にあった PSTD80 が制御レジスタ 7 の SSTD81 を

できる (E ビット = 1)。元の EAX は、プログラム戻りによって連係スタックから復元される。かくして、各プログラムはプログラムの呼出しに使用される ETE で指定された EAX を用いて実行することができる。一方、呼出し連係の間 EAX を不要にしておくことも可能であり (E ビット = 0)、それにより呼出しプログラムと同じアクセス権を被呼出しプログラムに持たせることができる。

制御レジスタ 3 の PKM77 をプログラム呼出しによって EKM に等しく設定することにより (M ビット = 1)、被呼出しプログラムは呼出しプログラムの PKM とは無関係の PKM を持つ。この結果、被呼出しプログラムは、それが設定できる PSW キー値の点から、呼出しプログラムよりも小さいアクセス権を持つことが可能になる。一方、新しい PKM77 は、所望であれば、古い PKM64 及び EKM のオア結合 (78 参照) に等しく設定されることもある (M ビット = 0)。

新しい SASN 及び新しい SSTD をそれぞれ

置き換える。スタッキング PC-ss が実行されかつ S ビットが 1 であれば、制御レジスタ 3 の SASN は新しい PASN によって置き換えられ、制御レジスタ 7 の SSTD は新しい PSTD によって置き換えられる。これらのオペレーションの後、ETE57 の ASN が 83 でテストされる。ASN が 0 であれば、PC-cp オペレーションの実行が完了したことを示す。しかし、ASN が 0 でなければ、PC-ss オペレーションが実行され、宛先空間に対する ASTE が得られる。

プログラム呼出しは、次の命令の保護されたコードを取り出すために、PSW キー 74 を EK で変更することでもできる (K ビット = 1)。

制御レジスタ 8 にある EAX を変更することにより (75 参照)、指名可能ユニットの作業を行なうように実行される各プログラムに対して、DUAL 及び PSAL にある ALE を使用する許可を異なった形で与えることができる。制御レジスタ 8 の EAX75 は、スタッキング・プログラム呼出しによって EEAX に等しく設定することが

新しい PASN 及び新しい PSTD に等しく設定すると (S = 1)、被呼出しプログラムが ALE T 値 X'00000001' の使用によって自動的に呼出しプログラムの一次アドレス空間のアクセス権を持つようになるのを阻止できる (アクセス権は ALE または DAS の二次 ASN 設定命令で得ることができる)。これは、被呼出しプログラムのアクセス権を呼出しプログラムのものより小さくする別の方法である。一方、新しい SASN79 及び新しい SSTD81 がそれぞれ古い PASN69 及び古い PSTD80 に等しく設定されることもある (S ビット = 0)。

第 17 図は、ASN 変換のステップの論理フローを示している。DAS 機構と同じく、プログラムを含む各アドレス空間は ASN を割り当てられ、その値は対応する ETE57 の 90 のところに書き込まれる。ASN90 は 2 つの数値 AFX91 及び ASX92 からなる。これも DAS 機構と同様である。制御レジスタ 14 は ASN 第 1 テーブル起点 (AFT0) 93 を含み、これは加算操作

94でAFX91と組み合わせると、ASN第1テーブルにおけるAFTE95の実アドレスを与える。AFTE95はASN第2テーブル起点(ASTO)98を含み、これは加算操作97でASX92と組み合わせると、ASN第2テーブル30におけるASTE98の実アドレスを与える。CRO.15が1の時は、ETE57にあるASTEアドレス100をASN変換の代わりに使用することができる。制御レジスタ14のビット12(CR14.12)はASN変換ビット(T101)である。これが0であれば、ASTEアドレス100もASN90も使用できない。CR14.12が1であれば、いずれかを使用できる。PC-ssオペレーションの場合、ASTE98のAX102及びETE57のASN90がそれぞれ制御レジスタ4の103及び104のところに置かれる。ASTE98のSTD105は制御レジスタ1の106のところに置かれる。CRO.15=1であれば、ASTEアドレスはPASTE0107として制御レジスタ5に置か

び25の一方に新しいALEを付加するのにも使用できる。サービス・プログラムは新しいアクセス・リスト・エントリを設定して、アクセス・レジスタ・モードのオペレーションで呼出しユーザが使用する新しいALETを供給することができる。アクセス・リスト・エントリを作成する時、呼出しユーザの制御レジスタ8からのEAXがALEAXとしてALE中に置かれる。ALEが作成されると、サービス・プログラムはそのALEに対するALETをユーザ・プログラムに戻す。ALETは、オペランドの取出しまたは記憶に使用するのに都合のよい形で、記憶させたり、他のアドレス空間へ送ったりすることができる。上述の許可手順は、許可されていないプログラムがALETを使用するのを阻止する。

アクセス・リスト・エントリの所有者は、アクセス・リスト・エントリの生成時にそれを私用エントリまたは公用エントリとして指定することができる。私用エントリの場合は、所有者または許可されたユーザだけがアドレス空間をアクセスで

れる。CRO.15=0であれば、制御レジスタ5にはASTE98のLTD108が置かれる。第17図のASN変換はDAS及びMASのいずれのオペレーションにも適用できる。

第15~17図に関連して説明したPC-ssオペレーションは、命令取出しのために制御を新しいアドレス空間へ移すのに使用できる。その場合、新しいアドレス空間は一次アドレス空間として設定される。たとえば、PC番号、エントリ・テーブル・エントリ及び連係テーブル・エントリがサービス提供者によって設定される時、そのPC番号を呼び出すプログラムのアクセス権を設定するためAKMが特定される。呼出しプログラムがエントリ・テーブル・エントリによって定義されたプログラムを呼び出す権限を持っていると(第16図のアンド操作63からわかる)、PCオペレーションは制御レジスタ8にあるEAXを変更し得る。

たとえば、PCオペレーションは、システム・サービスを呼び出して、アクセス・リスト24及

き、公用エントリの場合は、すべてのユーザがアクセスできる。公用エントリはPビット(ビット7)=0により示され、どのプログラムからも自由に使用できる。Pビットが1であれば、そのALEは許可されたプログラムしか使用できない。もし2以上のEAXがALEを使用できるのであれば、制御プログラムは、関連するアドレス空間のATにエントリを追加するための機構を提供する。

第18図は、プログラム許可検査付きのアクセス・レジスタ変換を示している。アクセス・レジスタ・オペレーションでALETを用いてオペランドの取出しまたは記憶を行なう場合、ALETが有効であることを確かめるため、ALETのビット0~6が115で調べられる。ALET中のPビット116が0であれば、アクセス・リストはDUALであり、Pビット116が1であれば、アクセス・リストはPSALである。DUALの場合、有効ALDは制御レジスタ2にアドレスが記憶されているDUCTから取り出される。PS

ALEの場合、有効ALDは制御レジスタ5にアドレスが記憶されている一次ASTE (PASTE)から取り出される。有効ALDは、アクセス・リスト起点及びアクセス・リスト長(ALL)を含む。ALENがアクセス・リストの外にないことを確かめるため、117でALEN及びALLが比較される。ALENがこの検査をパスすると、有効アクセス・リスト起点は加算操作119によりALENと結合され、アクセス・リスト121中のALE120のアドレスを生成する。まずALE120の無効ビット(ビット0)が121で検査される。無効ビットが0でALE120が有効であることを示している、124でALETのALESN122及びALE120のALESN123が比較される。ALESN122及びALESN123が等しければ、ALETはALE120のアクセスを許されており、ASTEアドレス125を用いてASTE126が取り出される。ASTE126の有効性は、128で無効ビット127を検査することにより確かめられる。A

確かめるため、EAX138及び許可権テーブル長(ATL)141が比較される。制御レジスタ8にあるEAX138は、ATO142によって起点が指定される許可権テーブルへの指標として使用される。許可権テーブル中のSビットがこのEAXに対して1にセットされていると、プログラムはアドレス空間のアクセスを許され、145でのDATオペレーションのためにSTD144が供給される。

アクセス・リスト・エントリ中の私用ビット及びALEAXフィールドは、ALEによって表わされるアドレス空間へのプログラムのアクセスを許可したり禁止したりする高性能の許可機構を提供する。私用ビットを0にしておくと、アクセス・リストを用いて実行されるすべてのプログラムがALEによって表わされるアドレス空間をアクセスできる。ALEの私用ビットを1にセットし、かつ制御レジスタ8にあるユーザのEAXをALEAXフィールドに等しくしておくと、特定のEAXを持ったプログラムがALEによって表わされ

STE126が有効であれば、ALE120がASTE126のアクセスを許されているかどうかを確かめるため、ASTESN130及びASTESN131が132で比較される。これらの検査により、ARTの有効性が確認される。

次に、呼出しプログラムのアドレス空間アクセス権が検査される。最初の検査は135で行なわれ、Pビット136が0かどうかを調べる。Pビット136が0であれば、すべてのプログラムがALEに関連するアドレス空間をアクセスすることができ、これ以上の検査は行なわれない。Pビット136が1であれば、ALEAX137及び制御レジスタ8にあるEAX138が比較器139で比較される。これらが等しければ、プログラムはこのアドレス空間のアクセスを許されており、これ以上の検査は行なわれない。139での比較が不一致であれば、ASN拡張許可検査が140で行なわれる。ASN拡張許可検査140では、制御レジスタ8にあるEAX138が許可権テーブルの外にあるエントリを指定していないことを

るアドレス空間をアクセスできる。最後に、ALEの私用ビットを1にセットし、かつ制御レジスタ8にあるユーザのEAXが目標空間の許可権テーブルにおいてSビットが1であるエントリを選択できるようにしておくと、異なるEAXで実行される複数のプログラムがALEによって表わされるアドレス空間をアクセスできる。

第19A図及び第19B図は、アクセス・レジスタ変換ステップ及び例外の流れを示している。ART論理が呼び込まれる時、アクセス・レジスタ0(ARO)が指定されたかどうか150で調べられる。アクセス・レジスタ0が指定されていると、ARTがテスト・アクセス・オペレーション(後述)によって呼び込まれたかどうか151で調べられる。アクセス・レジスタ0が指定されていないか、またはテスト・アクセス・オペレーションであったならば、アクセス・レジスタにあるALETの使用が152で示される。アクセス・レジスタ0が指定されかつこれがテスト・アクセス・オペレーションでなければ、153でX'0

0000000'がALETに割り当てられる。ついで、ALETがX'00000000'かどうか154で検査される。もし「はい」であれば、155で一次アドレス空間のためのSTDが制御レジスタ1から得られる。「いいえ」の場合は、156でALETがX'00000001'かどうか検査される。もし「はい」であれば、157で二次アドレス空間のためのSTDが制御レジスタから得られる。「いいえ」の場合は、158でALETのビット0~8が0かどうか検査される。ビット0~8が0でなければ、ALETの割り当てられた値は有効ではなく、159でALET指定例外が認識され、オペレーションは抑止される。

ビット0~8が0の場合は、160でALETのビット7が1かどうか検査される。1であれば、制御レジスタ5にあるPASTEOが161で解読され、PSALのための有効ALDが取り出される。ALETのビット7が0であれば、制御レジスタ2にあるDUCTOが162で解読さ

ば、172でALE順序例外が認識され、オペレーションは無効にされる。171で一致が検出されると、ALE中のASTEアドレスを用いて173でASTEが見つけれられる。ASTEアドレスが有効かどうか検査され、もし有効でなければ、174でアドレス指定例外が認識されて、オペレーションは抑止される。ASTEアドレスが有効であれば、ASTEが有効かどうかを見るため175でASTEの有効ビットが検査される。ASTEが有効でなければ、176でASTE有効性例外が認識され、オペレーションは無効にされる。ASTEが有効であれば、177でALEのASTESN及びASTEのASTESNが比較される。177での比較の結果が不一致であれば、178でASTE順序例外が認識され、オペレーションは無効にされる。

上述のステップ163~178は、得られたエントリが有効かどうかを決定するものである。次の179では、ALEの私用ビット(ビット7)が0かどうか検査される。また、ALEにある

れ、DUALのための有効ALDが取り出される。163で取出しアドレスが無効であることがわかると、164でアドレス指定例外が認識され、オペレーションは抑止される。アドレスが有効であれば、ALETのALENが有効ALL(有効ALDのビット25~31)の範囲外かどうか165で検査される。もし範囲外であれば、166でALEN変換例外が認識され、オペレーションは無効にされる。165での答えが「いいえ」であれば、167でALEが見つけれられ、ALEアドレスが有効かどうか検査される。ALEアドレスが有効でなければ、168でアドレス指定例外が認識され、オペレーションは抑止される。ALEアドレスが有効であれば、ALEが有効かどうかを見るため169でALEの有効ビットが検査される。ALEが有効でなければ、170でALEN変換例外が認識され、オペレーションは無効にされる。ALEが有効であれば、171でALETのALESN及びALEのALESNが比較される。171での比較の結果が不一致であれ

ALEX及び制御レジスタ8にあるEAXの比較も179で行なわれる。いずれかの検査で一致が検出されると、180でオペランドのためのSTDがアドレス空間のASTEから得られる。私用ビットが0であれば、プログラムはアクセスを許されており、アクセス・レジスタ変換の許可ステップは完了する。私用ビットが1であってもALEXがEAXと等しければ、プログラムはアクセスを許されており、アクセス・レジスタ変換の許可ステップは完了する。

プログラムが179ではまだ許可されていない場合は、181でASTEのビット30、31、60~63が0かどうかを見ることによりASTEの有効性が検査される。これらのビットが0でなければ、182でASN変換指定例外が認識され、オペレーションは抑止される。ASTEが有効であれば、EAXが許可権テーブルの外のエントリを指定していないことを確かめるため、183で制御レジスタ8にあるEAXビット0~11の値及び許可権テーブルの長さ(ATL)が比較

される。EAXビット0~11の値の方がATLよりも大きければ、184で拡張許可例外が認識され、オペレーションは抑止される。EAXが許可権テーブルの範囲内のエントリを指定していると、185で関連するエントリが許可権テーブルで見つけられる。この許可権テーブル・エントリのアドレスが有効でなければ、186でアドレス指定例外が認識され、オペレーションは抑止される。アドレスが有効であれば、185で見つけられた許可権テーブル・エントリの二次許可ビット(Sビット)が1かどうかを見る拡張許可検査が187で行なわれる。187での検査結果が「はい」であれば、呼出しプログラムはこのアドレス空間に関連する許可権テーブルによって許可されているプログラムのうちの1つであり、188でこのアドレス空間のためのSTDがASTEから得られる。187での検査結果が「いいえ」であれば、プログラムは許可されておらず、189で拡張許可例外が認識され、オペレーションは無効にされる。

2 = 指定されたALETは0または1ではなく、制御レジスタ5によってアドレスされるPSALにあり、指定されたEAXでのアクセスに対して有効である。

3 = 指定されたALETは1であるか、または指定されたEAXでのアクセスに対して無効である。

入力EAXを用いて許可例外についてALETをテストできるようにしておくと、プログラムは、ALETが呼出しプログラムのPASNを参照するのか(ALET=0)、DUALを参照するのか、呼出しプログラムのPSALを参照するのかを決定することができる。これは、プログラムをALETの内部フォーマットとは無関係にする。

TAR命令を用いてARTを実行する時、ALBエントリが作成される。かくして、ART中に例外が生じなければ、ARにあるALETが実際に使用される時、ALBはエントリを含む。

第20図はTAR命令の使用例を示している。指名可能ユニットのタスク制御ブロックTCB1

MAS機構は、前述のテスト・アクセス・オペレーションを実行するため、テスト・アクセス・レジスタ(TAR)命令を使用する。TAR命令のフォーマットは次のとおりである。

TAR A1, R2

TAR命令では、第1オペランドA1のアクセス・レジスタにあるものとして指定されたALETが、第2オペランドR2によって指定された汎用レジスタにあるEAXを用いてALET変換例外を検査される。TAR命令は、第19A図のステップ151で「はい」と判定されると、第19A図及び第19B図のARTオペレーションを実行させる。TAR命令は、次のテスト結果をPSWの条件コード(CC;第4図参照)に戻す。

0 = 指定されたALETは0であり、アクセスに対して有効である。

1 = 指定されたALETは0または1ではなく、制御レジスタ2によってアドレスされるDUALにあり、指定されたEAXでのアクセスに対して有効である。

は、200での実行中、5のEAXを持っている。このEAXにより、プログラムは、TCB1についての指名可能ユニット・アクセス・リスト(DUAL)の特定のエントリを使用することができる。201で、第1プログラムがアドレス空間ASN2にある第2プログラムに対しプログラム呼出しを行ない、第2プログラムで使用するべきALETを渡す。202で、ASN2にあるプログラムが8のEAXを用いて実行されるが、これは呼出しプログラムのEAXとは異なっている。もしプログラムが202で呼出しプログラムによって与えられたALETを使用すると、システム保全の問題が生じる。呼出しプログラムはALETを参照するEAX権限を持っていないことがあるが、ASN2のプログラムは持っている。ASN2のプログラムは、呼出しプログラムが渡されたALETを使用する権限を持っていたかどうかを調べる有効性検査を実行しなければならない。203で、ASN2のプログラムは、入力ALET及び呼出しプログラムのEAX(=5)を指定するT

AR命令を用いて、有効性検査を実行する。呼出しプログラムのEAXは、ASN2へのプログラム呼出しで作成された連係スタック・エントリから得られる。TAR命令の実行の結果、呼出しプログラムがALETの使用を許されていたことを示す条件コードが与えられると、ASN2のプログラムはその機能の遂行を続けることができる。呼出しプログラムがALETの使用を許されていないならば、ASN2のプログラムは呼出しプログラムを打ち切るか、または呼出しが成功しなかったことを示す戻りコードを呼出しプログラムに戻す。204で戻り命令により制御が戻されると、呼出しプログラムのEAX(=5)がスタックから復元され、ASN1のプログラムはこのEAXを用いて実行を続ける。

ALET有効性検査機能は頻りに要求される。第20図の例では、ASN2にあるプログラムへの呼出しのたびに必要とされる。この機能はオペレーティング・システムのサービス・ルーチンによっても提供できるが、そうするとオーバーヘッ

ドが大きくなり過ぎる。もしTAR機能がなければ、呼出しプログラムのALETを参照してEAXを変更しなければならないプログラムは、たとえば2つのPC命令を使用する必要がある。最初のPC命令はEAXを変更せず、呼出しプログラムのEAXを用いて呼出しプログラムのパラメータが参照される。あとで2番目のPC命令が実行され、被呼出しプログラムが使用する新しいEAXが与えられる。呼び出されたサービスは、その機能を遂行するのに異なったEAXを必要とすることがあり、この機構は正しいEAXの使用を可能にする。かくして、TAR機能はより効率のよい検査を可能にする。第20図の例では、TAR命令がDUALのALETと共に使用されているが、DUAL及びPSALのいずれのALETも使用できる。

F. 発明の効果

本発明により、プログラムに対してアドレス空間から命令を取り出したり1つ以上の他のいずれかのアドレス空間にオペランドを記憶したりする

ことを許可するアーキテクチャが達成された。

G. 用語解

AKM	許可キー・マーク
AL	アクセス・リスト・アドレス指定 ケイパビリティ・テーブル
AR	アクセス・レジスタ各ARはGPRと関連づけられる。
ART	アクセス・レジスタ変換-STDをARと関連づけられる方法。
AX	許可指標
ALB	ARTルックアサイド・バッファ ARTは、ARがBフィールド記憶域オペランド参照によってGPRで指定されるたびに生じ、ALBはその間の記憶域参照回数を減らす。
ALE	アクセス・リスト・エントリ
ALEAX	アクセス・リスト・エントリ許可指標。
ALEN	アクセス・リスト・エントリ番号-

	ALETのビット16~31が指定されたALEの番号である。
ALL	アクセス・リスト長-所定の数値として制御レジスタに記憶されており、最大1024個のALEまで可能である。
ALET	アクセス・リスト・エントリ・トークン-アクセス・リスト中の1つのエントリを指定する。
ALESN	アクセス・リスト・エントリ順序番号-ALET及びALEのビット8~15。
ASN	アドレス空間番号-アドレス空間を表わす。
ASTE	ASN第2テーブル・エントリ-従来の370/XAにおけるASTEを拡張したもので、無効ビット(I)及びSTDを含む。
ASTESN	ASTE順序番号-ALE及びASTEにあるASTESNが一致

AT L 許可権テーブル長
 D A S 二重アドレス空間
 D A S D 直接アクセス記憶装置
 D A T 動的アドレス変換—STDを用いて仮想アドレスを実アドレスに変換する。
 D U A L 指名可能ユニット・アクセス・リスト
 D U A L D D U A L 記述子—D U A L の起点（実アドレス）及び長さを含む。
 D U C T 指名可能ユニット制御テーブル—D U A L D 含み、C R 2 によって指定される。
 E A X 拡張許可指標
 E K M エントリ・キー・マスク
 E T E エントリ・テーブル・エントリ
 G P R (G R) 汎用レジスタオペランド及びアドレスを含む。
 L T D 連係テーブル記述子

M A S 多重アドレス空間
 P ビット (1) A L E T のビットで、D U A L または P S A L を選択する。
 (2) A L E 中のビットで、すべてのユーザがアクセスできるか（公用）または許可検査が必要か（私用）を示す。
 P A S T E 一次ASN第2テーブル・エントリ—P S A L D を含む。
 P C - c p 現一次に対するプログラム呼出し
 P C - s s 空間切替を伴うプログラム呼出し
 P E R プログラム事象記録
 P K M P S W キー・マスク
 P S A L 一次空間アクセス・リスト
 P S A L D P S A L 記述子—P A S T E に含まれ、P S A L の起点（実アドレス）及び長さからなる。
 P S T D 一次セグメント・テーブル記述子
 P S W プログラム状況ワード

S S T D 二次セグメント・テーブル記述子
 S T D セグメント・テーブル記述子

4. 図面の簡単な説明

第1図は本発明の概念を示すブロック図。

第2図はアクセス・レジスタ変換（A R T）の様子を示すブロック図。

第3図は本発明のMAS機構で使用される制御レジスタの構成を示す図。

第4図はMAS機構で使用されるPSWの内容を示す図。

第5図はアクセス・リスト・エントリ・トークン（A L E T）のフォーマットを示す図。

第6図はアクセス・リスト・エントリ（A L E）のフォーマットを示す図。

第7図は連係テーブル・エントリ（L T E）のフォーマットを示す図。

第8図はエントリ・テーブル・エントリ（E T E）のフォーマットを示す図。

第9図は連係スタックを示すブロック図。

第10図は連係スタックの状態エントリのフォー

マットを示す図。

第11図はASN第1テーブル・エントリ（A F T E）のフォーマットを示す図。

第12図はASN第2テーブル・エントリ（A S T E）のフォーマットを示す図。

第13図は許可権テーブル（A T）の構成を示す図。

第14図は指名可能ユニット制御テーブル（D U C T）のフォーマットを示す図。

第15図はPC番号変換の論理フローを示すブロック図。

第16図はスタッキング・オペレーションの論理フローを示すブロック図。

第17図はASN変換の論理フローを示すブロック図。

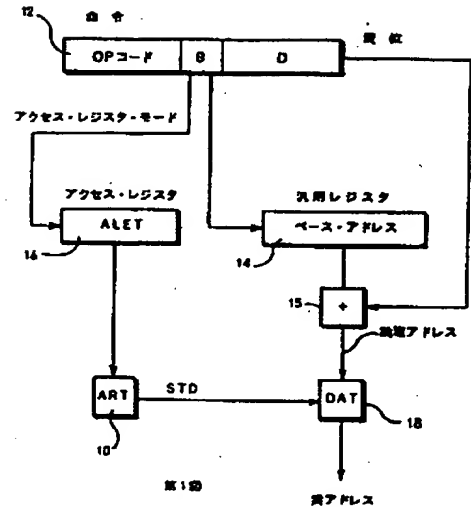
第18図はアクセス・レジスタ変換（A R T）の論理フローを示すブロック図。

第19A図及び第19B図はA R T オペレーションの流れ図。

第20図はテスト・アクセス・レジスタ命令の

実行の様子を示す図。

出願人 インターナショナル・ビジネス・
マシーンズ・コーポレーション
代理人 弁理士 頓 宮 孝 一
(外1名)



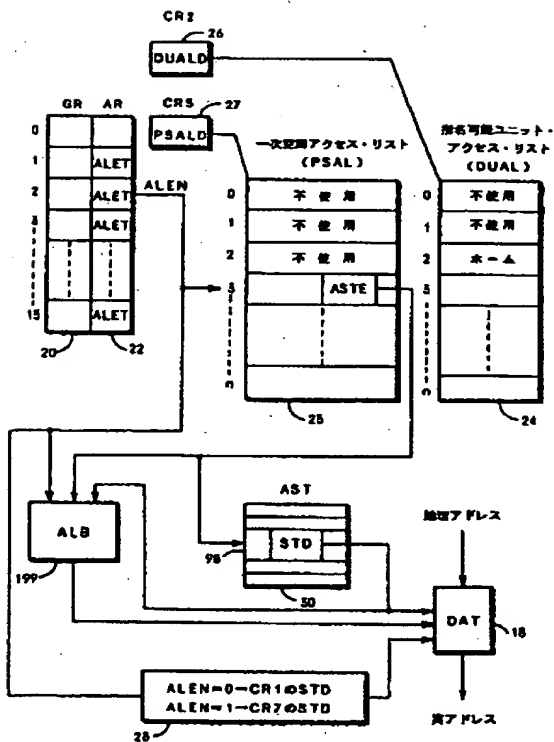
第1図

プログラム決定ワード (PSW)

O	R	O	O	O	T	O	E	K	E	M	W	P	M	O	D	C	C	プログラム	マスク	00000000
0					3	0		12		16		18		20		24		31		

A																				
32																				63

第4図



第2図

番号	制御レジスタ
0	一次セグメント・テーブル記号 (PSTO)
1	PSTO 20 PSTL 25
2	指名可能ユニット制御テーブル記号 (DUCTO) 24
3	PSWキー・マスク (PKM) 15 16 二次アドレス範囲番号 (SASN) 31
4	許可制御 (AX) 15 16 一次アドレス範囲番号 (FASN) 31
5	一次ASTE範囲 (PASTE) 26
6	二次セグメント・テーブル記号 (SSTO)
7	SSTO 20 SSTL 25
8	拡張許可制御 (EAX) 15
9	
10	
11	
12	
13	ネーム・セグメント・テーブル記号 (NSTO)
14	NSTO 20 NSTL 25
15	ASM第1テーブル記号 (ASTO) 20 (AFTO) 31
16	退避スタック・エントリ・アドレス 26

第3図

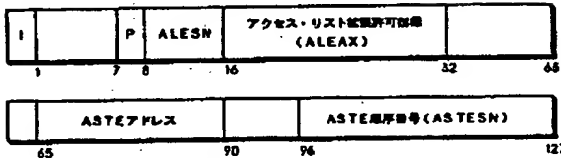
第5図

アクセス・リスト・エントリ・トークン(ALET)



第6図

アクセス・リスト・エントリ(ALE)



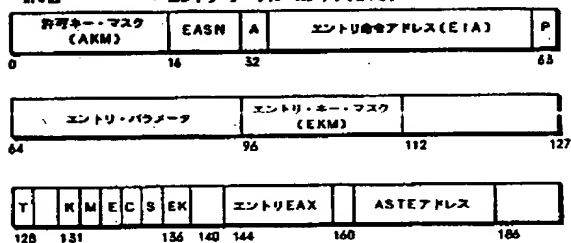
第7図

拡張テーブル・エントリ(LTE)



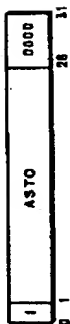
第8図

エントリ・テーブル・エントリ(ETE)



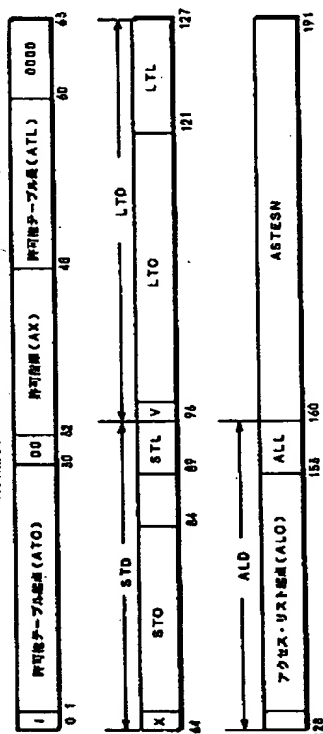
第11図

拡張1テーブル・エントリ(AFTE)



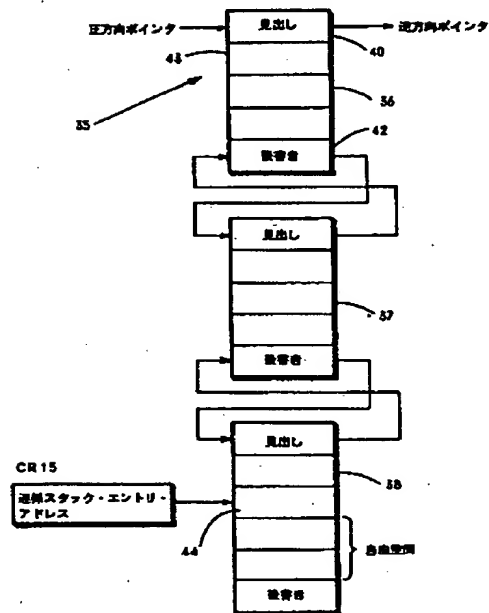
第12図

拡張2テーブル・エントリ(ASTE)



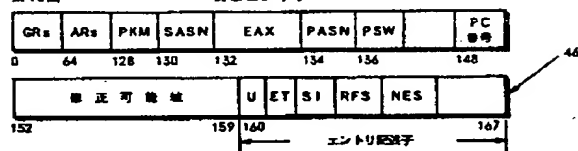
第9図

逆順スタック(LS)



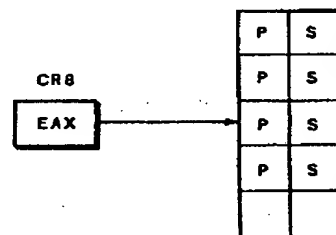
第10図

拡張エントリ



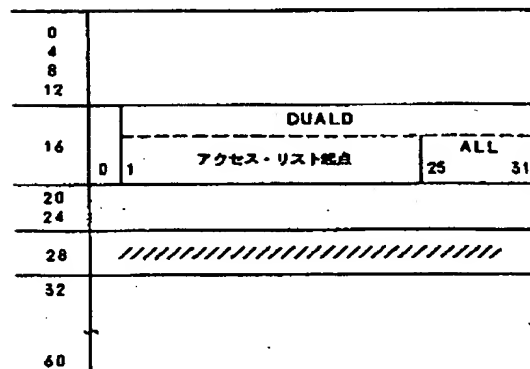
第13図

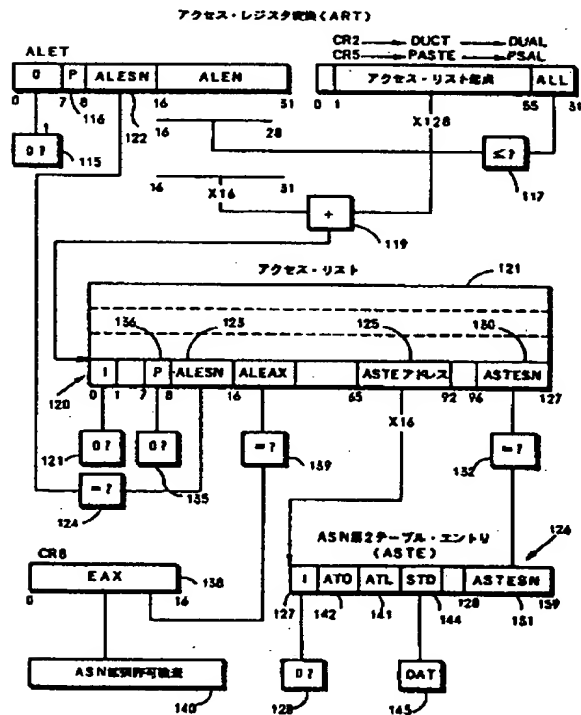
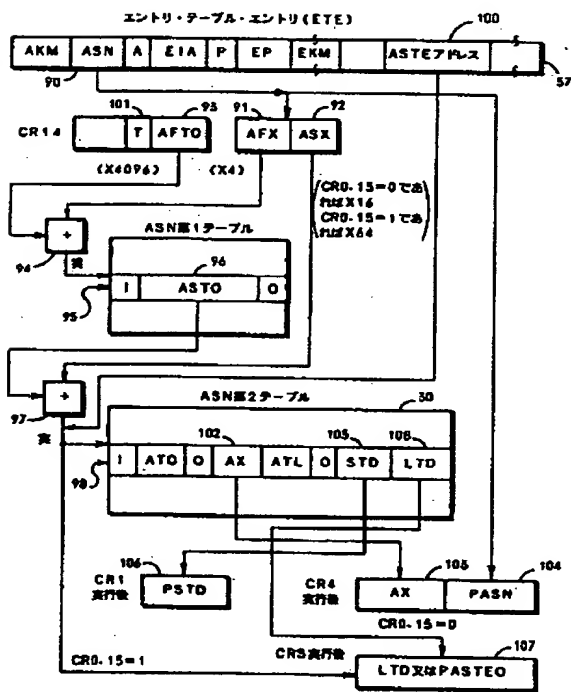
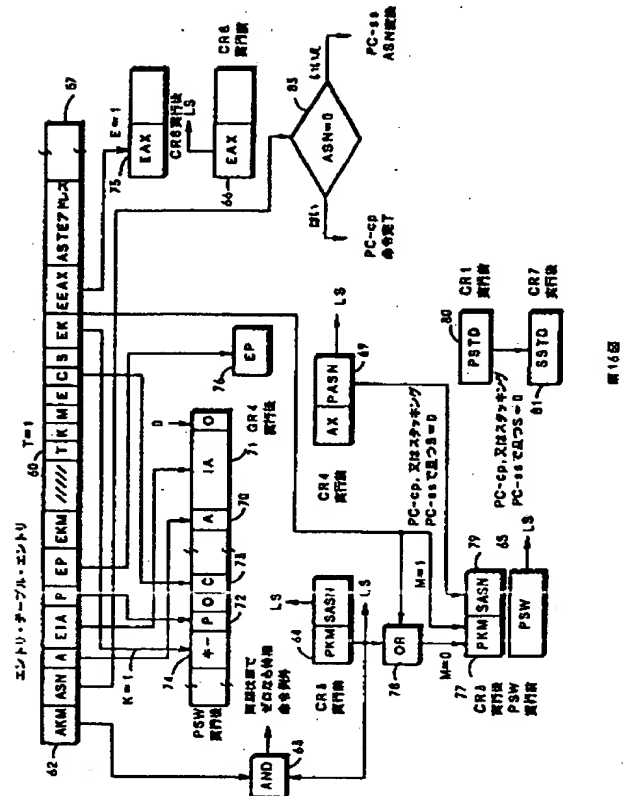
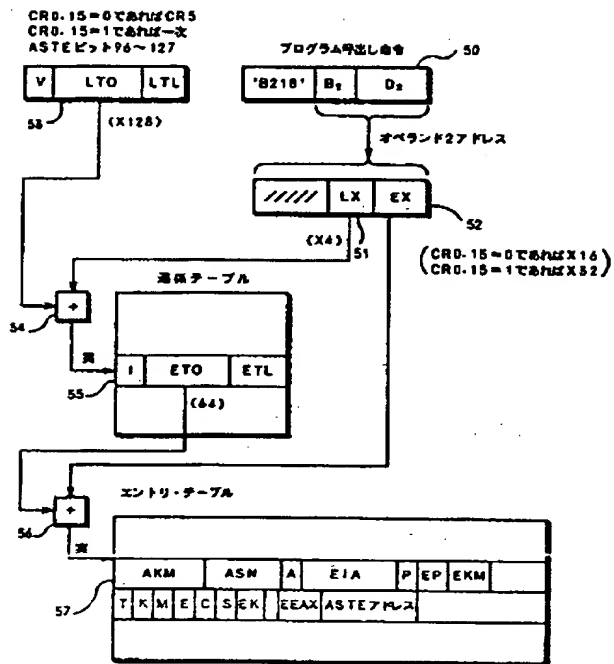
許可テーブル(AT)



第14図

拡張名可能ユニット制御テーブル(DUCT)





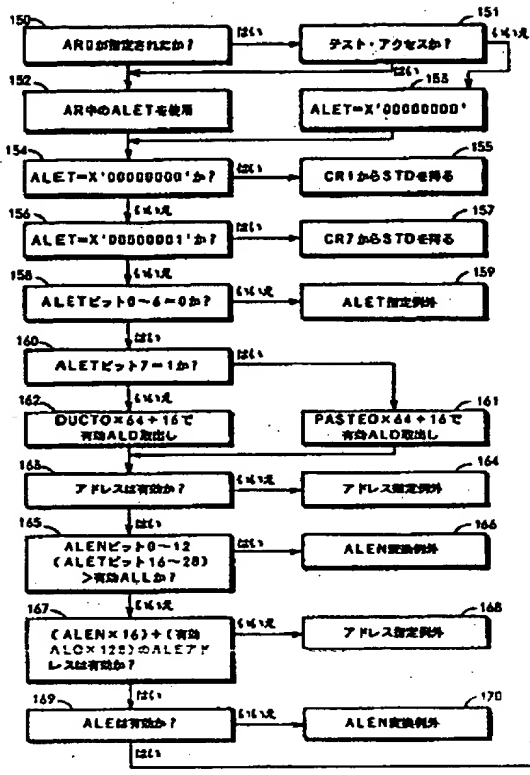


図19A

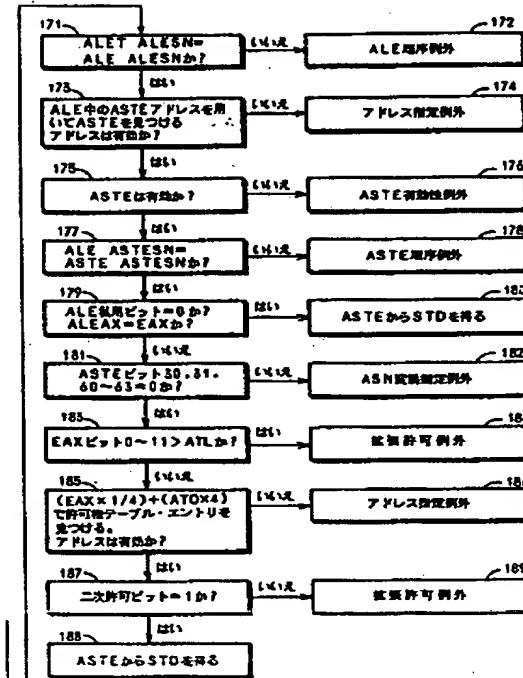


図19B

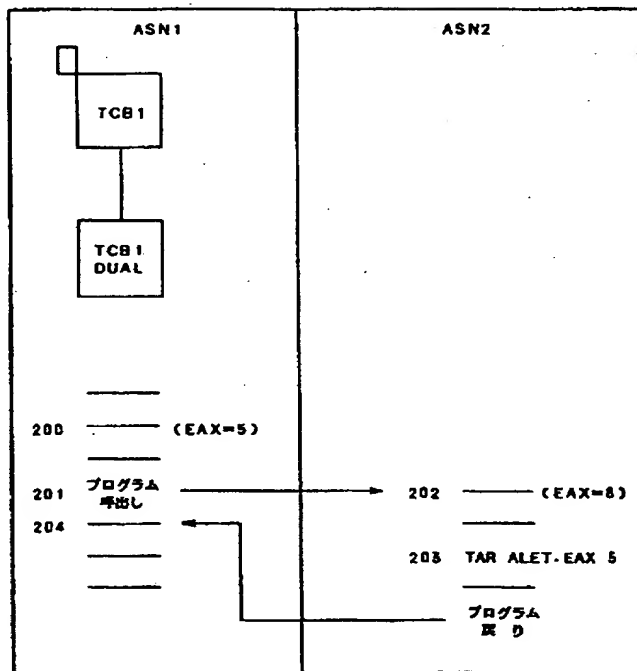


図20

第1頁の続き

⑦発明者	ジヤステイン・ラル フ・バットウエル	アメリカ合衆国ニューヨーク州ミルトン、チエストナツ ト・レーン81-5番地
⑧発明者	カール・エドワード・ クラーク	アメリカ合衆国ニューヨーク州ボキブシイ、パート・ドラ イブ46番地
⑨発明者	アレン・ジョージ・ガ ネク	アメリカ合衆国ニューヨーク州チャバク、ピン・オーク・ レーン18番地
⑩発明者	ジェームズ・ラム	アメリカ合衆国カリフォルニア州レッドウッド・シテイ、 ケイブ・ハタラス・コート34番地
⑪発明者	マイケル・ジェラー ド・マール	アメリカ合衆国ニューヨーク州ラグランジヴイレ、スクウ エア・ヴツズ・ドライブ20番地
⑫発明者	デヴィド・リチャー ド・ページ	イギリス国ハンプシャー、ラムズイ、シイカモアー・クロ ス13番地
⑬発明者	ケネス・アーネスト・ ブラムベック	アメリカ合衆国ニューヨーク州ボキブシイ、デイシイ・レ ーン7番地
⑭発明者	キヤスパー・アンソニ イ・スカルズイ	アメリカ合衆国ニューヨーク州ボキブシイ、アパートメン ト7イー、アカデミイ・ストリート160番地
⑮発明者	リチャード・ジョン・ シヤマルズ	アメリカ合衆国ニューヨーク州ワツピンガーズ・フォール ズ、エツヂ・ヒル・ドライブ7番地